

Optimal Assembly for High Throughput Shotgun Sequencing*

Guy Bresler, Ma'ayan Bresler, and David Tse[†]

Dept. of EECS, UC Berkeley. Email: {gbresler,mbresler,dtse}@eecs.berkeley.edu

Abstract

We present a framework for the design of optimal assembly algorithms for shotgun sequencing under the criterion of complete reconstruction. We derive a lower bound on the read length and the coverage depth required for reconstruction in terms of the repeat statistics of the genome. Building on earlier works, we design a de Bruijn graph based assembly algorithm which can achieve very close to the lower bound for repeat statistics of a wide range of sequenced genomes, including the GAGE datasets. The results are based on a set of necessary and sufficient conditions on the DNA sequence and the reads for reconstruction. The conditions can be viewed as the shotgun sequencing analogue of Ukkonen-Pevzner's necessary and sufficient conditions for Sequencing by Hybridization.

*The authors thank Yun Song, Lior Pachter, Sharon Aviran, and Serafim Batzoglou for stimulating discussions. This work is supported by the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370. M. Bresler is also supported by NSF grant DBI-0846015.

[†]Author names are in alphabetical order.

1 Introduction

1.1 Problem statement

DNA sequencing is the basic workhorse of modern day biology and medicine. Since the sequencing of the Human Reference Genome ten years ago, there has been an explosive advance in sequencing technology, resulting in several orders of magnitude increase in throughput and decrease in cost. Multiple “next-generation” sequencing platforms have emerged. All of them are based on the whole-genome shotgun sequencing method, which entails two steps. First, many short reads are extracted from random locations on the DNA sequence, with the length, number, and error rates of the reads depending on the particular sequencing platform. Second, the reads are assembled to reconstruct the original DNA sequence.

Assembly of the reads is a major algorithmic challenge, and over the years dozens of assembly algorithms have been proposed to solve this problem [28]. Nevertheless, the assembly problem is far from solved, and it is not clear how to compare algorithms nor where improvement might be possible. The difficulty of comparing algorithms is evidenced by the recent assembly evaluations Assemblathon 1 [2] and GAGE [23], where which assembler is “best” depends on the particular dataset as well as the performance metric used. In part this is a consequence of metrics for partial assemblies: there is an inherent tradeoff between larger contiguous fragments (contigs) and fewer mistakes in merging contigs (misjoins). But more fundamentally, independent of the metric, performance depends critically on the dataset, i.e. length, number, and quality of the reads, as well as the complexity of the genome sequence. With an eye towards the near future, we seek to understand the interplay between these factors by using the intuitive and unambiguous metric of *complete* reconstruction¹. Note that this objective of reconstruct-

ing the original DNA sequence from the reads contrasts with the many *optimization-based* formulations of assembly, such as shortest common superstring (SCS) [7], maximum-likelihood [16], [11], and various graph-based formulations [22], [14]. When solving one of these alternative formulations, there is no guarantee that the optimal solution is indeed the original sequence.

Given the goal of complete reconstruction, the most basic questions are 1) **feasibility**: given a set of reads, is it *possible* to reconstruct the original sequence? 2) **optimality**: which *algorithms* can successfully reconstruct whenever it is feasible to reconstruct? The feasibility question is a measure of the intrinsic *information* each read provides about the DNA sequence, and for given sequence statistics depends on characteristics of the sequencing technology such as read length and noise statistics. As such, it can provide an algorithm-independent basis for evaluating the efficiency of a sequencing technology. Equally important, algorithms can be evaluated on their relative read length and data requirements, and compared against the fundamental limit.

In studying these questions, we consider the most basic shotgun sequencing model where N noiseless reads² of a fixed length L base pairs are uniformly and independently drawn from a DNA sequence of length G . In this statistical model, feasibility is rephrased as the question of whether, for given sequence statistics, the correct sequence can be reconstructed with probability $1 - \epsilon$ when N reads of length L are sampled from the genome. We note that answering the feasibility question of whether each N, L pair is sufficient to reconstruct is equivalent to finding the minimum required N (or the *coverage depth* $c = NL/G$) as a function of L .

A lower bound on the minimum coverage depth needed was obtained by Lander and Waterman [9]. Their lower bound $c_{\text{LW}} = c_{\text{LW}}(L, \epsilon)$ is the minimum number of randomly located reads needed to cover the entire DNA sequence with a given target success probability $1 - \epsilon$. While this is clearly a necessary condition, it is

to be represented by a contiguous sequence.

²Reads are thus exact subsequences of the DNA.

¹The notion of complete reconstruction can be thought of as a mathematical idealization of the notion of “finishing” a sequencing project as defined by the National Human Genome Research Institute [18], where finishing a chromosome requires at least 95% of the chromosome

in general not tight: only requiring the reads to cover the entire genome sequence does not guarantee that consecutive reads can actually be stitched back together to recover the original sequence. Characterizing when the reads can be reliably stitched together, i.e. determining feasibility, is an open problem. In fact, the ability to reconstruct depends crucially on the *repeat statistics* of the DNA sequence.

An earlier work [13] has answered the feasibility and optimality questions under an i.i.d. model for the DNA sequence. However, real DNA, especially those of eukaryotes, have much longer and complex repeat structures. Here, we are interested in determining feasibility and optimality given *arbitrary* repeat statistics. This allows us to evaluate algorithms on statistics from already sequenced genomes, and gives confidence in predicting whether the algorithms will be useful for an *unseen* genome with similar statistics.

1.2 Results

Our approach results in a pipeline, which takes as input a genome sequence and desired success probability $1 - \epsilon$, computes a few simple repeat statistics, and from these statistics computes a feasibility plot that indicates for which L, N reconstruction is possible. Fig. 1 displays the simplest of the statistics, the number of repeats as a function of the repeat length ℓ . Fig. 2 shows the resulting feasibility plot produced for the statistics of human chromosome 19 (henceforth hc19) with success probability 99%. The horizontal axis signifies read length L and the vertical axis signifies the normalized coverage depth $\bar{c} := c/c_{\text{LW}}$, the coverage depth c normalized by c_{LW} , the coverage depth required as per Lander-Waterman [9] in order to cover the sequence.

Since the coverage depth must satisfy $c \geq c_{\text{LW}}$, the normalized coverage depth satisfies $\bar{c} \geq 1$, and we plot the horizontal line $\bar{c} = 1$. This lower bound holds for *any* assembly algorithm. In addition, there is another lower bound, shown as the thick black nearly vertical line in Fig. 2. In contrast to the coverage lower bound, this lower bound is a function of the repeat statistics. It has a vertical asymptote at $L_{\text{crit}} :=$

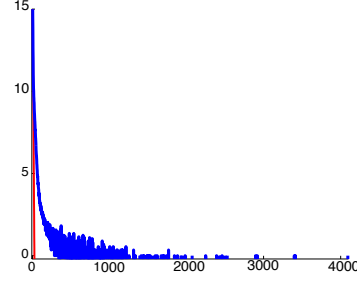


Figure 1: For hc19, a log plot of number of repeats as a function of the repeat length ℓ . Red line is what would have been predicted by an i.i.d. fit.

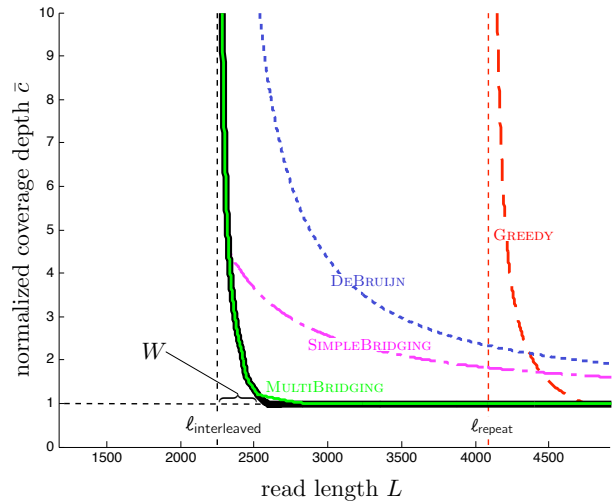


Figure 2: Thick black lines are lower bounds on feasibility which holds for all algorithms, and colored curves are performance achieved by specific algorithms. Four such curves are shown: the greedy algorithm and three de Bruijn graph based algorithms.

$\max\{\ell_{\text{interleaved}}, \ell_{\text{triple}}\} + 1$, where $\ell_{\text{interleaved}}$ is the length of the longest interleaved repeats in the DNA sequence and ℓ_{triple} is the length of the longest triple repeat (see Section 2 for precise definitions). Our lower bound can be viewed as a generalization of a result of Ukkonen [26] for Sequencing by Hybridization to the shotgun sequencing setting.

Each colored curve in the feasibility plot is the lower boundary of the set of feasible N, L pairs for a specific algorithm. The rightmost curve is the one achieved by the greedy algorithm, which merges reads with largest overlaps first (used for example in TIGR [25], CAP3 [5], and more recently SSAKE [27]). As seen in Fig. 2, its per-

formance curve asymptotes at $L = \ell_{\text{repeat}}$, the length of the longest repeat. De Bruijn graph based algorithms (e.g. [6] and [22]) take a more global view via the construction of a de Bruijn graph out of all the K-mers of the reads. The performance curves of all K-mer graph based algorithms asymptote at read length $L = L_{\text{crit}}$, but different algorithms use read information in a variety of ways to resolve repeats in the K-mer graph and thus have different coverage depth requirement beyond read length L_{crit} . By combining the ideas from several existing algorithms (including [22], [19]) we designed MULTIBRIDGING, which is very close to the lower bound for this dataset. Thus Fig. 2 answers, up to a very small gap, the feasibility of assembly for the repeat statistics of hc19, where successful reconstruction is desired with probability 99%.

We produce similar plots for a dozen or so datasets (see supplementary material). For datasets where $\ell_{\text{interleaved}}$ is significantly larger than ℓ_{triple} (the majority of the datasets we looked at, including those used in the recent GAGE assembly algorithm evaluation [23]), MULTIBRIDGING is near optimal, thus allowing us to characterize the fundamental limits for these repeat statistics (Fig. 9). On the other hand, if ℓ_{triple} is close to or larger than $\ell_{\text{interleaved}}$, there is a gap between the performance of MULTIBRIDGING and the lower bound (see for example Fig. 3). The reason for the gap is explained in Section 3.4.

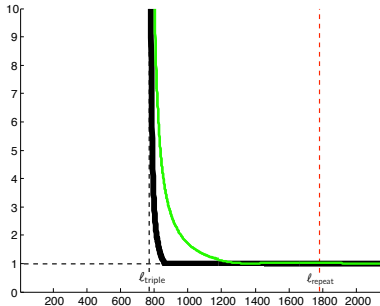


Figure 3: Performance of MULTIBRIDGING on *P. Marinus*, where $\ell_{\text{triple}} > \ell_{\text{interleaved}}$.

An interesting feature of the feasibility plots is that for typical repeat statistics exhibited by DNA data, the minimum coverage depth is characterized by a *critical phenomenon*: If the read

length L is below $L_{\text{crit}} = \ell_{\text{interleaved}}$, reliable reconstruction of the DNA sequence is impossible no matter what the coverage depth is, but if the read length L is slightly above L_{crit} , then covering the sequence suffices, i.e. $\bar{c} = c/c_{\text{LW}} = 1$. The sharpness of the critical phenomenon is described by the size of the *critical window*, which refers to the range of L over which the transition from one regime to the other occurs. For the case when MULTIBRIDGING is near optimal, the width W of the window size can be well approximated as:

$$W \approx \frac{L_{\text{crit}}}{2r + 1}, \quad \text{where } r := \frac{\log \frac{G}{L_{\text{crit}}}}{\log \epsilon^{-1}}. \quad (1)$$

For the hc19 dataset, the critical window size evaluates to about 19% of L_{crit} .

In Sections 2 and 3, we discuss the underlying analysis and algorithm design supporting the plots. The curves are all computed from formulas, which are validated by simulations in Section 4. We return in Section 5 to put our contributions in a broader perspective and discuss extensions to the basic framework. All proofs can be found in the appendix.

2 Lower bounds

In this section we discuss lower bounds, due to coverage analysis and certain repeat patterns, on the required coverage depth and read length. The style of analysis here is continued in Section 3, in which we search for an assembly algorithm that performs close to the lower bounds.

2.1 Coverage bound

Lander and Waterman's coverage analysis [9] gives the well known condition for the number of reads N_{LW} required to cover the entire DNA sequence with probability at least $1 - \epsilon$. In the regime when $L \ll G$, one may make the standard assumption that the starting locations of the N reads follow a Poisson process with rate $\lambda = N/G$, and the number N_{LW} is to a very good approximation given by the solution to the equation

$$N_{\text{LW}} = \frac{G}{L} \log \frac{N_{\text{LW}}}{\epsilon}. \quad (2)$$

The corresponding coverage depth is $c_{\text{LW}} = N_{\text{LW}}L/G$. This is our baseline coverage depth against which to compare the coverage depth of various algorithms. For each algorithm, we will plot

$$\bar{c} := \frac{c}{c_{\text{LW}}} = \frac{N}{N_{\text{LW}}},$$

the coverage depth required by that algorithm normalized by c_{LW} . Note that \bar{c} is also the ratio of the number of reads N required by an algorithm to N_{LW} . The requirement $\bar{c} \geq 1$ is due to the lower bound on the number of reads obtained by the Lander-Waterman coverage condition.

2.2 Ukkonen's condition

A second constraint on reads arises from repeats. A lower bound on the read length L follows from Ukkonen's condition [26]: if there are *interleaved repeats* or *triple repeats* in the sequence of length at least $L - 1$, then the likelihood of observing the reads is the same for more than one possible DNA sequence and hence correct reconstruction is not possible. Fig. 4 shows an example with interleaved repeats. (Note that we assume $1 - \epsilon > 1/2$, so random guessing between equally likely sequences is not viable.)

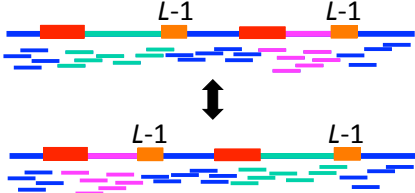


Figure 4: The likelihood of observing the reads under two possible sequences (the green and magenta segments swapped) is the same. Here, the two red subsequences form a repeat and the two orange subsequences form another repeat.

We take a moment to carefully define the various types of repeats. Let \mathbf{s}_t^ℓ denote the length- ℓ subsequence of the DNA sequence \mathbf{s} starting at position t . A *repeat* of length ℓ is a subsequence appearing twice, at some positions t_1, t_2 (so $\mathbf{s}_{t_1}^\ell = \mathbf{s}_{t_2}^\ell$) that is maximal (i.e. $s(t_1 - 1) \neq s(t_2 - 1)$ and $s(t_1 + \ell) \neq s(t_2 + \ell)$). Similarly, a *triple repeat* of length ℓ is a subsequence appearing three times, at positions t_1, t_2, t_3 , such that $\mathbf{s}_{t_1}^\ell = \mathbf{s}_{t_2}^\ell = \mathbf{s}_{t_3}^\ell$,

and such that neither of $s(t_1 - 1) = s(t_2 - 1) = s(t_3 - 1)$ nor $s(t_1 + \ell) = s(t_2 + \ell) = s(t_3 + \ell)$ holds³. A *copy* is a single one of the instances of the subsequence's appearances. A *pair* of repeats refers to two repeats, each having two copies. A pair of repeats, one at positions t_1, t_3 with $t_1 < t_3$ and the second at positions t_2, t_4 with $t_2 < t_4$, is *interleaved* if $t_1 < t_2 < t_3 < t_4$ or $t_2 < t_1 < t_4 < t_3$ (Fig. 4). The length of a pair of interleaved repeats is defined to be the length of the shorter of the two repeats.

Ukkonen's condition implies a lower bound on the read length,

$$L > L_{\text{crit}} := \max\{\ell_{\text{interleaved}}, \ell_{\text{triple}}\} + 1.$$

Here $\ell_{\text{interleaved}}$ is the length of the longest pair of interleaved repeats on the DNA sequence and ℓ_{triple} is the length of the longest triple repeat.

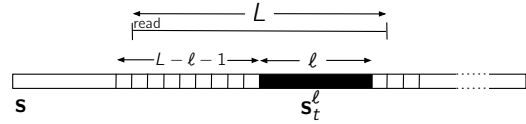


Figure 5: A subsequence \mathbf{s}_t^ℓ is bridged if and only if there exists at least one read which covers at least one base on both sides of the subsequence, i.e. the read arrives in the preceding length $L - \ell - 1$ interval.

Ukkonen's condition says that for read lengths less than L_{crit} , reconstruction is impossible no matter what the coverage depth is. But it can be generalized to provide a lower bound on the coverage depth for read lengths greater than L_{crit} , through the important concept of *bridging* as shown in Figure 5. We observe that in Ukkonen's interleaved or triple repeats, the actual length of the repeated subsequences is irrelevant; rather, to cause confusion it is enough that all the copies of the pertinent repeats are unbridged. This leads to the following theorem.

Theorem 1. *Given a DNA sequence \mathbf{s} and a set of reads, if there is a pair of interleaved repeats or a triple repeat whose copies are all unbridged, then there is another sequence \mathbf{s}' of the same length under which the likelihood of observing the reads is the same.*

³Note that a subsequence that is repeated f times gives rise to $\binom{f}{2}$ repeats and $\binom{f}{3}$ triple repeats.

For brevity, we will call a repeat or a triple repeat *bridged* if at least one copy of the repeat is bridged, and a pair of interleaved repeats *bridged* if at least one of the repeats is bridged. Thus, the above theorem says that a necessary condition for reconstruction is that all interleaved and triple repeats are bridged.

How does Theorem 1 imply a lower bound on the coverage depth? Focus on the longest pair of interleaved repeats and suppose the read length L is between the lengths of the shorter and the longer repeats. The probability this pair is unbridged is $(p_{\ell_{\text{interleaved}}}^{\text{unbridged}})^2$, where

$$p_{\ell}^{\text{unbridged}} := \mathbb{P}[\ell\text{-length subseq. is unbridged}] = e^{\frac{N}{G}(L-\ell-1)^+}. \quad (3)$$

Theorem 1 implies that the probability of making an error in the reconstruction is at least $1/2$ if this event occurs. Hence, the requirement that $P_{\text{error}} \leq \epsilon$ implies a lower bound on the number of reads N :

$$N \geq \frac{G}{(L - \ell_{\text{interleaved}} - 1) \ln(1/(2\epsilon))}. \quad (4)$$

A similar lower bound can be derived using the longest triple repeat. A slightly tighter lower bound can be obtained by taking into consideration the bridging of *all* the interleaved and triple repeats, not only the longest one, resulting in the black curve in Fig. 2.

3 Towards optimal assembly

We now begin our search for algorithms performing close to the lower bounds derived in the previous section. Algorithm assessment begins with obtaining deterministic sufficient conditions for success in terms of repeat-bridging. We then find the necessary N and L in order to satisfy these sufficient conditions with a target probability $1 - \epsilon$. The required coverage depth for each algorithm depends only on certain repeat statistics extracted from the DNA data, which may be thought of as *sufficient statistics*.

3.1 Greedy algorithm

The greedy algorithm, denoted GREEDY, with pseudocode in section C.1, is described as fol-

lows. Starting with the initial set of reads, the two fragments (i.e. subsequences) with maximum length overlap are merged, and this operation is repeated until a single fragment remains. Here the overlap of two fragments \mathbf{x}, \mathbf{y} is a suffix of \mathbf{x} equal to a prefix of \mathbf{y} , and merging two fragments results in a single longer fragment.

Theorem 2. GREEDY reconstructs the original sequence \mathbf{s} if every repeat is bridged.

Theorem 2 allows us to determine the coverage depth required by GREEDY: we must ensure that all repeats are bridged. By the union bound,

$$\mathbb{P}[\text{some repeat is unbridged}] \leq \sum_m a_m (p_m^{\text{unbridged}})^2, \quad (5)$$

where $p_m^{\text{unbridged}}$ is defined in (3) and a_m is the number of repeats of length m . Setting the right-hand side of (5) to ϵ ensures $P_{\text{error}} \leq \epsilon$ and yields the performance curve of GREEDY in Fig. 2. Note that the repeat statistics $\{a_m\}$ are sufficient to compute this curve.

GREEDY requires $L > \ell_{\text{repeat}} + 1$, whereas the lower bound has its asymptote at $L = \ell_{\text{interleaved}} + 1$. In chromosome 19, for instance, there is a large difference between $\ell_{\text{interleaved}} = 2248$ and $\ell_{\text{repeat}} = 4092$, and in Fig 2 we see a correspondingly large gap. GREEDY is evidently sub-optimal in handling interleaved repeats. Its strength, however, is that once the reads are slightly longer than ℓ_{repeat} , coverage of the sequence is sufficient for correct reconstruction. Thus if $\ell_{\text{repeat}} \approx \ell_{\text{interleaved}}$, then GREEDY is close to optimal.

3.2 K -mer algorithms

The greedy algorithm fails when there are unbridged repeats, even if there are no unbridged *interleaved* repeats, and therefore requires a read length much longer than that required by Ukkonen's condition. As we will see, K -mer algorithms do not have this limitation.

3.2.1 Background

In the introduction we mention Sequencing By Hybridization (SBH), for which Ukkonen's con-

dition was originally introduced. In the SBH setting, an optimal algorithm matching Ukkonen’s condition is known, due to Pevzner [21].

Pevzner’s algorithm is based on finding an appropriate cycle in a K -mer graph (also known as a de Bruijn graph) with $K = L - 1$ (see e.g. [1] for an overview). A K -mer graph is formed by first creating a node in the graph for each unique K -mer (length K subsequence) in the set of reads, and then adding an edge with overlap $K - 1$ between any two nodes representing K -mers that are *adjacent* in a read, i.e. offset by a single nucleotide. Edges thus correspond to unique $(K + 1)$ -mers in \mathbf{s} and paths correspond to longer subsequences obtained by merging the constituent nodes. There exists a cycle corresponding to the original sequence \mathbf{s} , and reconstruction entails finding this cycle.

As is common, we will replace edges corresponding to an unambiguous path by a single node (c.f. Fig. 6). Since the subsequences at some nodes are now longer than K , this is no longer a K -mer graph, and we call the more general graph a sequence graph. The simplified graph is called the *condensed sequence graph*.

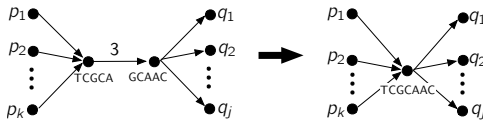


Figure 6: Contracting an edge by merging the incident nodes. Repeating this operation results in the condensed graph.

The condensed graph has the useful property that if the original sequence \mathbf{s} is reconstructible, then \mathbf{s} is determined by a unique Eulerian cycle:

Theorem 3. *Let \mathbb{G}_0 be the K -mer graph constructed from the $(K + 1)$ -spectrum \mathcal{S}_{K+1} of \mathbf{s} , and let \mathbb{G} be the condensed sequence graph obtained from \mathbb{G}_0 . If Ukkonen’s condition is satisfied, i.e. there are no triple or interleaved repeats of length at least K , then there is a unique Eulerian cycle \mathcal{C} in \mathbb{G} and \mathcal{C} corresponds to \mathbf{s} .*

Theorem 3 characterizes, deterministically, the values of K for which reconstruction from the $(K + 1)$ -spectrum is possible. We proceed with application of the K -mer graph approach to shotgun sequencing data.

3.2.2 Basic K -mer algorithm

Starting with Idury and Waterman [6], and then Pevzner et al.’s [22] EULER algorithm, most current assembly algorithms for shotgun sequencing are based on the K -mer graph. Idury and Waterman [6] made the key observation that SBH with subsequences of length $K + 1$ can be *emulated* by shotgun sequencing if each read overlaps the subsequent read by K : the set of all $(K + 1)$ -mers within the reads is equal to the $(K + 1)$ -spectrum \mathcal{S}_{K+1} . The resultant algorithm DEBRUIJN which consists of constructing the K -mer graph from the $(K + 1)$ -spectrum observed in the reads, condensing the graph, and then identifying an Eulerian cycle, has sufficient conditions for correct reconstruction as follows.

Theorem 4. *DEBRUIJN with parameter choice K reconstructs the original sequence \mathbf{s} if:*

- (a) $K > \ell_{\text{interleaved}}$
- (b) $K > \ell_{\text{triple}}$
- (c) *adjacent reads overlap by at least K*

Lander and Waterman’s coverage analysis applies also to Condition (c) of Theorem 4, yielding a normalized coverage depth requirement $\bar{c} = 1/(1 - K/L)$. The larger the overlap K , the higher the coverage depth required. Conditions (a) and (b) say that the smallest K one can choose is $K = \max\{\ell_{\text{triple}}, \ell_{\text{interleaved}}\} + 1$, so

$$\bar{c} = \frac{1}{1 - \frac{\max\{\ell_{\text{triple}}, \ell_{\text{interleaved}}\} + 1}{L}}. \quad (6)$$

The performance of DEBRUIJN is plotted in Fig. 2. DEBRUIJN significantly improves on GREEDY by obtaining the correct first order performance: given sufficiently many reads, the read length L may be decreased to $\max\{\ell_{\text{triple}}, \ell_{\text{interleaved}}\} + 1$. Still, the number of reads required to approach this critical length is far above the lower bound. The following subsection pursues reducing K in order to reduce the required number of reads.

3.3 Improved K -mer algorithms

Algorithm DEBRUIJN ignores a lot of information contained in the reads, and indeed all of

the K -mer based algorithms proposed by the sequencing community (including [6], [22], [24], [4], [10], [29]) use the read information to a greater extent than the naive DEBRUIJN algorithm. Better use of the read information, as described below in algorithms SIMPLEBRIDGING and MULTIBRIDGING, will allow us to relax the condition $K > \max\{\ell_{\text{interleaved}}, \ell_{\text{triple}}\}$ for success of DEBRUIJN, which in turn reduces the high coverage depth required by Condition (c).

Existing algorithms use read information in a variety of distinct ways to resolve repeats. For instance, Pevzner et al. [22] observe that for graphs where each edge has multiplicity one, if one copy of a repeat is bridged, the repeat can be resolved through what they call a “detachment”. The algorithm SIMPLEBRIDGING described below is very similar, and resolves repeats with two copies if at least one copy is bridged.

Meanwhile, other algorithms are better suited to higher edge multiplicities due to higher order repeats; IDBA (Iterative DeBruijn Assembler) [19] creates a series of K -mer graphs, each with larger K , and at each step uses not just the reads to identify adjacent K -mers, but also all the unbridged paths in the K -mer graph with smaller K . Although not stated explicitly in their paper, we observe here that if all copies of every repeat are bridged, then IDBA correctly reconstructs.

However, it is suboptimal to require that *all* copies of every repeat up to the maximal K be bridged. We introduce MULTIBRIDGING, which combines the aforementioned ideas to simultaneously allow for single-bridged double repeats, triple repeats in which all copies are bridged, and unbridged non-interleaved repeats.

3.3.1 SIMPLEBRIDGING

SIMPLEBRIDGING improves on DEBRUIJN by resolving bridged 2-repeats (i.e. a repeat with exactly two copies in which at least one copy is bridged by a read). Condition (a) $K > \ell_{\text{interleaved}}$ for success of DEBRUIJN (ensuring that no interleaved repeats appear in the initial K -mer graph) is updated to require only no *unbridged* interleaved repeats, which matches the lower bound. With this change, Condition (b) $K > \ell_{\text{triple}}$

forms the bottleneck for typical DNA sequences. Thus SIMPLEBRIDGING is optimal with respect to interleaved repeats, but it is suboptimal with respect to triple repeats.

SIMPLEBRIDGING deals with repeats by performing surgery on certain nodes in the sequence graph. In the sequence graph, a repeat corresponds to a node we call an X -node, a node with in-degree and out-degree each at least two (e.g. Fig. 7). A self-loop adds one each to the in-degree and out-degree. The cycle $\mathcal{C}(\mathbf{s})$ traverses each X -node at least twice, so X -nodes correspond to repeats in \mathbf{s} . We call an X -node traversed exactly twice a 2- X -node; these nodes correspond to 2-repeats, and are said to be bridged if the corresponding repeat in \mathbf{s} is bridged.

In the repeat resolution step of SIMPLEBRIDGING (illustrated in Fig. 7), bridged 2- X -nodes are duplicated in the graph and incoming and outgoing edges are inferred using the bridging read, reducing possible ambiguity.

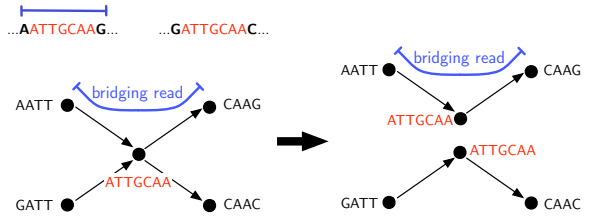


Figure 7: An example of the bridging step in SIMPLEBRIDGING.

Theorem 5. SIMPLEBRIDGING with parameter choice K reconstructs the original sequence \mathbf{s} if:

- (a) all interleaved repeats are bridged
- (b) $K > \ell_{\text{triple}}$
- (c) adjacent reads overlap by at least K .

By the union bound,

$$\begin{aligned} & \mathbb{P}[\text{some interleaved repeat is unbridged}] \\ & \leq \sum_{m,n} b_{m,n} \left(p_m^{\text{unbridged}}\right)^2 \left(p_n^{\text{unbridged}}\right)^2 \end{aligned} \quad (7)$$

where $b_{m,n}$ is the number of interleaved repeats in which one repeat is of length m and the other is of length n . To ensure that condition (a) in the above theorem fails with probability no more than ϵ , the right hand side of (7) is set to be ϵ ; this imposes a constraint on the coverage depth. Furthermore, conditions (b) and

(c) imply that the normalized coverage depth $\bar{c} \geq 1/(1 - (\ell_{\text{triple}} + 1)/L)$. These two constraints together yield the performance curve of SIMPLEBRIDGING in Figure 2.

3.3.2 MULTIBRIDGING

We now turn to triple repeats. As previously observed, it can be challenging to resolve repeats with more than one copy [22], because an edge into the repeat may be paired with more than one outgoing edge. As discussed above, our approach here shares elements with IDBA [19]: we note that increasing the node length serves to resolve repeats. Unlike IDBA, we do not increase the node length globally.

As noted in the previous subsection, repeats correspond to nodes in the sequence graph we call *X-nodes*. Here the converse is false: not all repeats correspond to X-nodes. A repeat is said to be *all-bridged* if *all* repeat copies are bridged, and an X-node is called all-bridged if the corresponding repeat is all-bridged.

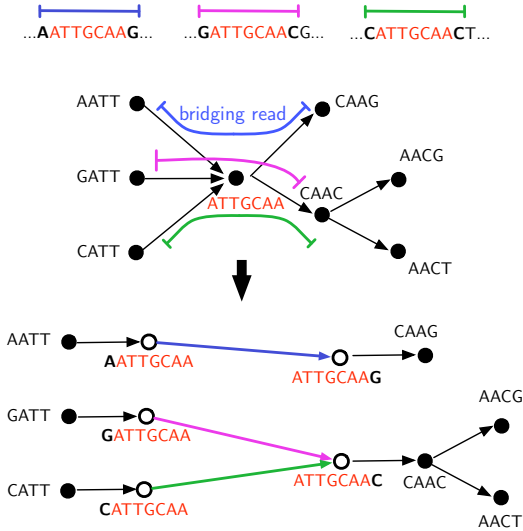


Figure 8: MULTIBRIDGING resolves an X-node with label ATTGCAA corresponding to a triple repeat.

The requirement that triple repeats be all-bridged allows them to be resolved *locally* (Fig. 8). The X-node resolution procedure given in Step 4 of MULTIBRIDGING can be interpreted in the *K*-mer graph framework as increasing *K* locally so that repeats do not appear in the graph. In order to do this, we introduce the following notation for extending nodes: Given

an edge (\mathbf{v}, \mathbf{q}) with weight $a_{\mathbf{v}, \mathbf{q}}$, let $\mathbf{v} \rightarrow \mathbf{q}$ denote \mathbf{v} extended one base to the right along (\mathbf{v}, \mathbf{q}) , i.e. $\mathbf{v} \rightarrow \mathbf{q} = \mathbf{v} \mathbf{q}_{a_{\mathbf{v}, \mathbf{q}} + 1}^1$ (notation introduced in Sec. 2.2). Similarly, let $\mathbf{p} \rightarrow \mathbf{v} = \mathbf{p}_{\text{end} - a_{\mathbf{p}, \mathbf{v}}}^1 \mathbf{v}$. MULTIBRIDGING is described as follows.

Algorithm 1 MULTIBRIDGING. Input: reads \mathcal{R} , parameter K . Output: sequence $\hat{\mathbf{s}}$.

K-mer steps 1-3:

1. For each subsequence \mathbf{x} of length K in a read, form a node with label \mathbf{x} .
 2. For each read, add edges between nodes representing adjacent *K*-mers in the read.
 3. Condense the graph (c.f. Fig. 6).
 4. *Bridging step:* (See Fig. 8). While there exists a bridged X-node \mathbf{v} : (i) For each edge $(\mathbf{p}_i, \mathbf{v})$ with weight $a_{\mathbf{p}_i, \mathbf{v}}$, create a new node $\mathbf{u}_i = \mathbf{p}_i \rightarrow \mathbf{v}$ and an edge $(\mathbf{p}_i, \mathbf{u}_i)$ with weight $1 + a_{\mathbf{p}_i, \mathbf{v}}$. Similarly for each edge $(\mathbf{v}, \mathbf{q}_j)$, create a new node $\mathbf{w}_j = \mathbf{v} \rightarrow \mathbf{q}_j$ and edge $(\mathbf{w}_j, \mathbf{q}_j)$. (ii) If \mathbf{v} has a self-loop (\mathbf{v}, \mathbf{v}) with weight $a_{\mathbf{v}, \mathbf{v}}$, add an edge $(\mathbf{v} \rightarrow \mathbf{v}, \mathbf{v} \rightarrow \mathbf{v})$ with weight $a_{\mathbf{v}, \mathbf{v}} + 2$. (iii) Remove node \mathbf{v} and all incident edges. (iv) For each pair $\mathbf{u}_i, \mathbf{w}_j$ adjacent in a read, add edge $(\mathbf{u}_i, \mathbf{w}_j)$. If exactly one each of the \mathbf{u}_i and \mathbf{w}_j nodes have no added edge, add the edge. (v) Condense graph.
 5. *Finishing step:* Find an Eulerian cycle in the graph and return the corresponding sequence.
-

Theorem 6. *The algorithm MULTIBRIDGING reconstructs the sequence \mathbf{s} if:*

- (a) *all interleaved repeats are bridged*
- (b) *all triple repeats are **all-bridged***
- (c) *the sequence is covered by the reads.*

A similar analysis as for SIMPLEBRIDGING yields the performance curve of MULTIBRIDGING in Figure 2.

3.4 Gap to lower bound

The only difference between the sufficient condition guaranteeing the success of MULTIBRIDGING and the necessary condition of the lower bound is the bridging condition of *triple* repeats: while MULTIBRIDGING requires bridging *all three copies* of the triple repeats, the necessary condition requires only bridging *a single copy*. When ℓ_{triple} is significantly smaller

than $\ell_{\text{interleaved}}$, the bridging requirement of interleaved repeats dominates over that of triple repeats and MULTIBRIDGING achieves very close to the lower bound. This occurs in hc19 and the majority of the datasets we looked at. (See Fig. 9 and the plots in the supplementary material.) A critical phenomenon occurs as L increases: for $L < L_{\text{crit}}$ reconstruction is impossible, over a small critical window the bridging requirement of interleaved repeats (primarily the longest) dominates, and then for larger L , coverage suffices.

On the other hand, when ℓ_{triple} is comparable or larger than $\ell_{\text{interleaved}}$, then MULTIBRIDGING has a gap in the coverage depth to the lower bound (see for example Fig. 3). If we further assume that the longest triple repeat is dominant, then this gap can be calculated to be a factor of $3 \cdot \frac{\log 3\epsilon^{-1}}{\log \epsilon^{-1}} \approx 3.72$ for $\epsilon = 10^{-2}$. This gap occurs only within the critical window where the repeat-bridging constraint is active. Beyond the critical window, the coverage constraint dominates and MULTIBRIDGING is optimal. Further details are provided in the appendices.

4 Simulations and complexity

In order to verify performance predictions, we implemented and ran the algorithms on simulated error-free reads from sequenced genomes. For each algorithm, we sampled (N, L) points predicted to give $< 5\%$ error, and recorded the number of times correct reconstruction was achieved out of 100 trials. Fig. 9 shows results for the three GAGE reference sequences.

We now estimate the run-time of MULTIBRIDGING. The algorithm has two phases: the K -mer graph formation step, and the repeat resolution step. The K -mer graph formation runtime can be easily bounded by $O((L-K)NK)$, assuming $O(K)$ look-up time for each of the $(L-K)N$ K -mers observed in reads. This step is common to all K -mer graph based algorithms, so previous works to decrease the practical runtime or memory requirements are applicable.

The repeat resolution step depends on the repeat statistics and choice of K . It can be loosely bounded as $O\left(\sum_{\ell=K}^L L \sum_{\text{of length } \ell}^{\text{max repeats } x} d_x\right)$.

The second sum is over distinct maximal repeats x of length ℓ and d_x is the number of (not necessarily maximal) copies of repeat x . The bound comes from the fact that each maximal repeat of length $K < \ell < L$ is resolved via exactly one bridged X-node, and each such resolution requires examining at most the Ld_x distinct reads that contain the repeat. We note that $\sum_{\ell=K}^L L \sum_{\text{of length } \ell}^{\text{max repeats } x} d_x < L \sum_{\ell=K}^L a_\ell$, and the latter quantity is easily computable from our sufficient statistics.

For our data sets, with appropriate choice of K , the bridging step is much simpler than the K -mer graph formation step: for *R. sphaeroides* we use $K = 40$ to get $\sum_{\ell=K}^L La_\ell = 412$; in contrast, $N > 22421$ for the relevant range of L . Similarly, for hc14, using $K = 300$, $\sum_{\ell=K}^L La_\ell = 661$ while $N > 733550$; for *S. Aureus*, $\sum_{\ell=K}^L La_\ell = 558$ while $N > 8031$.

5 Discussions and extensions

The notion of *optimal shotgun assembly* is not commonly discussed in the literature. One reason is that there is no universally agreed-upon metric of success. Another reason is that most of the optimization-based formulations of assembly have been shown to be NP-hard, including Shortest Common Superstring [3], [7], De Bruijn Superwalk [22], [12], and Minimum s-Walk on the string graph [14], [12]. Thus, it would seem that optimal assembly algorithms are out of the question from a computational perspective. What we show in this paper is that if the goal is complete reconstruction, then one can define a clear notion of optimality, and moreover there is a computationally efficient assembly algorithm (MULTIBRIDGING) that is near optimal for a wide range of DNA repeat statistics. So while the reconstruction problem may well be NP-hard, typical instances of the problem seem much easier than the worst-case, a possibility already suggested by Nagarajan and Pop [17].

The MULTIBRIDGING algorithm is near optimal in the sense that, for a wide range of repeat statistics, it requires the minimum read length and minimum coverage depth to achieve com-

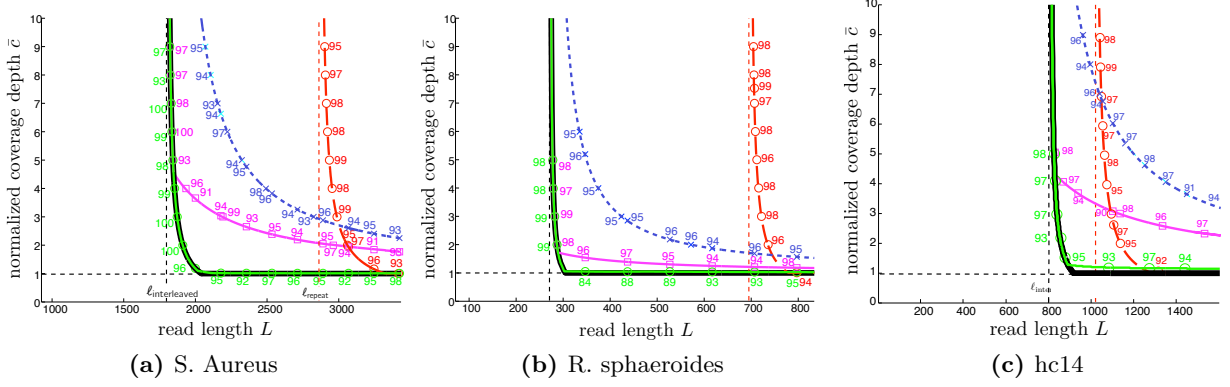


Figure 9: Simulation results for each of the GAGE reference genomes. Each simulated (N, L) point is marked with the number of correct reconstructions (e.g. 93, 98, 95) on 100 simulated read sets. All four algorithms (GREEDY, DEBRUIJN, SIMPLEBRIDGING, and MULTIBRIDGING) were run on *S. Aureus*, *R. sphaeroides* and hc14. Note that MULTIBRIDGING is very close to the lower bound on all 3 datasets.

plete reconstruction. However, since the repeat statistics of a genome to be sequenced are usually not known in advance, this minimum required read length and minimum required coverage depth may also not be known in advance. In this context, it would be useful for the MULTIBRIDGING algorithm to *validate* whether its assembly is correct. More generally, an interesting question is to seek algorithms which are not only optimal in their data requirements but also provide a measure of confidence in their assemblies.

How realistic is the goal of complete reconstruction given current-day sequencing technologies? The minimum read lengths L_{crit} required for complete reconstruction on the datasets we examined are typically on the order of 500 – 3000 base pairs (bp). This is substantially longer than the reads produced by Illumina, the current dominant sequencing technology, which produces reads of lengths 100-200bp; however, other technologies produce longer reads. PacBio reads can be as long as several thousand base pairs, and as demonstrated by [8], the noise can be cleaned by Illumina reads to enable near-complete reconstruction. Thus our framework is already relevant to some of the current cutting edge technologies. To make our framework more relevant to short-read technologies such as Illumina, an important direction is to incorporate mate-pairs in the read model, which can help to resolve long repeats with short reads. Other extensions to the basic shotgun sequencing model:

heterogenous read lengths: This occurs in some technologies where the read length is random (e.g. Pacbio) or when reads from multiple technologies are used. Generalized Ukkonen’s conditions and the sufficient conditions of MULTIBRIDGING extend verbatim to this case, and only the computation of the bridging probability (3) has to be slightly modified.

non-uniform read coverage: Again, only the computation of the bridging probability has to be modified. One issue of interest is to investigate whether reads are sampled less frequently from long repeat regions. If so, our framework can quantify the performance hit.

double strand: DNA is double-stranded and consists of a length- G sequence \mathbf{u} and its reverse complement $\bar{\mathbf{u}}$. Each read is either sampled from \mathbf{u} or $\bar{\mathbf{u}}$. This more realistic scenario can be mapped into our single-strand model by defining \mathbf{s} as the length- $2G$ concatenation of \mathbf{u} and $\bar{\mathbf{u}}$, transforming each read into itself and its reverse complement so that there are $2N$ reads. Generalized Ukkonen’s conditions hold verbatim for this problem, and MULTIBRIDGING can be applied, with the slight modification that instead of looking for a single Eulerian path, it should look for two Eulerian paths, one for each component of the sequence graph after repeat-resolution. An interesting aspect of this model is that, in addition to interleaved repeats on the single strand \mathbf{u} , *reverse complement repeats* on \mathbf{u} will also induce interleaved repeats on the sequence \mathbf{s} .

References

- [1] P. Compeau, P. Pevzner, and G. Tesler, *How to apply de Bruijn graphs to genome assembly*, Nat Biotech **29** (2011), no. 11, 987–991.
- [2] D. Earl, K. Bradnam, J.S. John, A. Darling, D. Lin, J. Fass, H.O.K. Yu, V. Buffalo, D.R. Zerbino, M. Diekhans, et al., *Assemblathon 1: A competitive assessment of de novo short read assembly methods*, Genome research **21** (2011), no. 12, 2224–2241.
- [3] J. Gallant, D. Maier, and J. Astorer, *On finding minimal length superstrings*, Journal of Computer and System Sciences **20** (1980), no. 1, 50–58.
- [4] Sante Gnerre, Iain MacCallum, Dariusz Przybylski, Filipe J. Ribeiro, Joshua N. Burton, Bruce J. Walker, Ted Sharpe, Giles Hall, Terrance P. Shea, Sean Sykes, Aaron M. Berlin, Daniel Aird, Maura Costello, Riza Daza, Louise Williams, Robert Nicol, Andreas Gnirke, Chad Nusbaum, Eric S. Lander, and David B. Jaffe, *High-quality draft assemblies of mammalian genomes from massively parallel sequence data*, Proceedings of the National Academy of Sciences **108** (2011), no. 4, 1513–1518.
- [5] X Huang and A Madan, *CAP3: A DNA sequence assembly program*, Genome Research **9** (1999), no. 9, 868–877.
- [6] R. Idury and M.S. Waterman, *A new algorithm for DNA sequence assembly*, J. Comp. Bio **2** (1995), 291–306.
- [7] John D. Kececioğlu and Eugene W. Myers, *Combinatorial algorithms for DNA sequence assembly*, Algorithmica **13** (1993), 7–51.
- [8] Sergey Koren, Michael C Schatz, Brian P Walenz, Jeffrey Martin, Jason T Howard, Ganeshkumar Ganapathy, Zhong Wang, David A Rasko, W Richard McCombie, Erich D Jarvis, and Adam M Phillippy, *Hybrid error correction and de novo assembly of single-molecule sequencing reads*, Nat Biotech **30** (2012), 693–700.
- [9] E.S. Lander and M.S. Waterman, *Genomic mapping by fingerprinting random clones: A mathematical analysis*, Genomics **2** (1988), no. 3, 231–239.
- [10] Iain Maccallum, Dariusz Przybylski, Sante Gnerre, Joshua Burton, Ilya Shlyakhter, Andreas Gnirke, Joel Malek, Kevin McKernan, Swati Ranade, Terrance P Shea, Louise Williams, Sarah Young, Chad Nusbaum, and David B Jaffe, *Allpaths 2: small genomes assembled accurately and with high continuity from short paired reads*, Genome Biol **10** (2009), no. 10, R103.
- [11] P. Medvedev and M. Brudno, *Maximum likelihood genome assembly*, Journal of computational Biology **16** (2009), no. 8, 1101–1116.
- [12] P. Medvedev, K. Georgiou, G. Myers, and M. Brudno, *Computability of models for sequence assembly*, Algorithms in Bioinformatics (2007), 289–301.
- [13] S.A. Motahari, G. Bresler, and D. Tse, *Information theory of DNA sequencing*, 2012, <http://arxiv.org/abs/1203.6233>.
- [14] E. Myers, *The fragment assembly string graph*, Bioinformatics **21** (2005), ii79–ii85.
- [15] Eugene W. Myers, Granger G. Sutton, Art L. Delcher, Ian M. Dew, Dan P. Fasulo, Michael J. Flanigan, Saul A. Kravitz, Clark M. Mobarry, Knut H. J. Reinert, Karin A. Remington, Eric L. Anson, Randall A. Bolanos, Hui-Hsien Chou, Catherine M. Jordan, Aaron L. Halpern, Stefano Lonardi, Ellen M. Beasley, Rhonda C. Brandon, Lin Chen, Patrick J. Dunn, Zhongwu Lai, Yong Liang, Deborah R. Nusskern, Ming Zhan, Qing Zhang, Xiangqun Zheng, Gerald M. Rubin, Mark D. Adams, and J. Craig Venter, *A whole-genome assembly*

- of drosophila*, Science **287** (2000), no. 5461, 2196–2204.
- [16] E.W. Myers, *Toward simplifying and accurately formulating fragment assembly*, Journal of Computational Biology **2** (1995), no. 2, 275–290.
- [17] N. Nagarajan and M. Pop, *Parametric complexity of sequence assembly: theory and applications to next generation sequencing*, Journal of computational biology **16** (2009), no. 7, 897–908.
- [18] NIH National Human Genome Research Institute, *Human genome sequence quality standards*, Dec 2012, [Online; accessed Dec-12-2012] <http://www.genome.gov/10000923>.
- [19] Y. Peng, H. Leung, S. Yiu, and F. Chin, *IDBA—a practical iterative de Bruijn graph de novo assembler*, Research in Computational Molecular Biology, Springer, 2010, pp. 426–440.
- [20] P. Pevzner, *ℓ -tuple DNA sequencing: computer analysis*, J Biomol Struct Dyn. **7** (1989), no. 1, 63–73.
- [21] P. A. Pevzner, *DNA physical mapping and alternating Eulerian cycles in colored graphs*, Algorithmica **13** (1995), no. 1/2, 77–105.
- [22] P. A. Pevzner, H. Tang, and M. S. Waterman, *An Eulerian path approach to DNA fragment assembly*, Proc Natl Acad Sci USA **98** (2001), 9748–53.
- [23] Steven L. Salzberg, Adam M. Phillippy, Aleksey Zimin, Daniela Puiu, Tanja Magoc, Sergey Koren, Todd J. Treangen, Michael C. Schatz, Arthur L. Delcher, Michael Roberts, Guillaume Marcais, Mihai Pop, and James A. Yorke, *GAGE: A critical evaluation of genome assemblies and assembly algorithms*, Genome research **22** (2012), no. 3, 557–567.
- [24] Jared T. Simpson, Kim Wong, Shaun D. Jackman, Jacqueline E. Schein, Steven J.M. Jones, and Inanc Birol, *ABYSS: A parallel assembler for short read sequence data*, Genome Research **19** (2009), no. 6, 1117–1123.
- [25] G. G. Sutton, O. White, M. D. Adams, and Ar Kerlavage, *TIGR Assembler: A new tool for assembling large shotgun sequencing projects*, Genome Science & Technology **1** (1995), 9–19.
- [26] E. Ukkonen, *Approximate string matching with q -grams and maximal matches*, Theoretical Computer Science **92** (1992), no. 1, 191–211.
- [27] R.L. Warren, G.G. Sutton, S.J. Jones, and R.A. Holt, *Assembling millions of short DNA sequences using SSAKE*, Bioinformatics **23** (2007), 500–501.
- [28] Wikipedia, *Sequence assembly — Wikipedia, the free encyclopedia*, 2012, [Online; accessed Nov-20-2012] http://en.wikipedia.org/wiki/Sequence_assembly.
- [29] Daniel R Zerbino and Ewan Birney, *Velvet: algorithms for de novo short read assembly using de Bruijn graphs*, Genome Res **18** (2008), no. 5, 821–9.

A Supplementary Material

In this supplementary material, we display the output of our pipeline for 9 datasets (in addition to hc19, whose output is in the introduction, and the GAGE datasets *R. sphaeroides*, *S. Aureus*, and hc14). For each dataset we plot

$$\log(1 + a_\ell),$$

the log of one plus the number of repeats of each length ℓ . From the repeat statistics a_m , $b_{m,n}$, and c_m , we produce a feasibility plot. The thick black line denotes the lower bound on feasible N, L , and the green line is the performance achieved by MULTIBRIDGING.

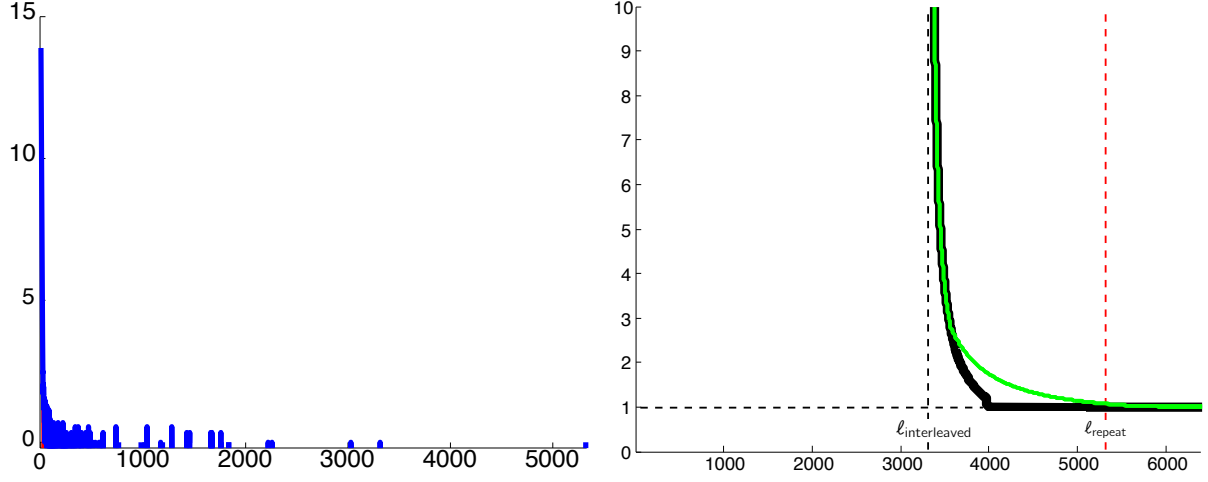


Figure 10: Lactofidus. $G = 2,078,001$, $\ell_{\text{triple}} = 3027$, $\ell_{\text{interleaved}} = 3313$, $\ell_{\text{repeat}} = 5321$.

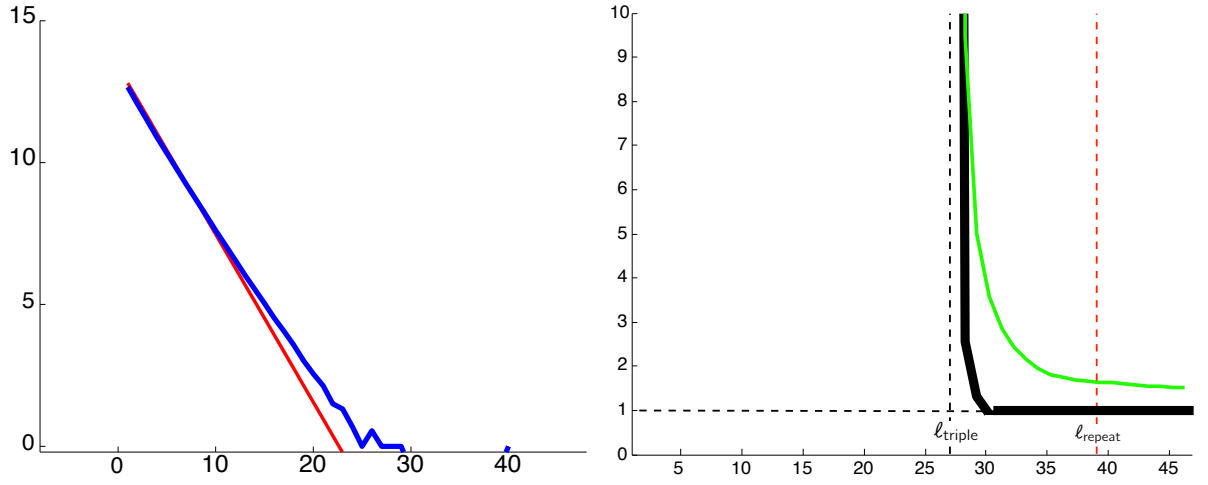


Figure 11: Buchnera. $G = 642,122$, $\ell_{\text{triple}} = 27$, $\ell_{\text{interleaved}} = 23$, $\ell_{\text{repeat}} = 39$.

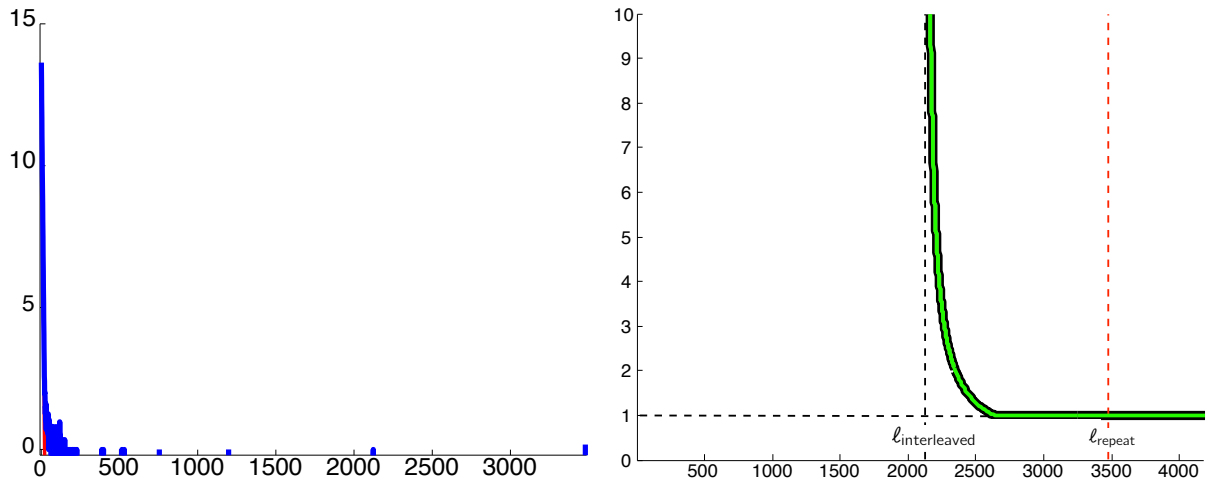


Figure 12: Heli51. $G = 1,589,954$, $\ell_{\text{triple}} = 219$, $\ell_{\text{interleaved}} = 2122$, $\ell_{\text{repeat}} = 3478$.

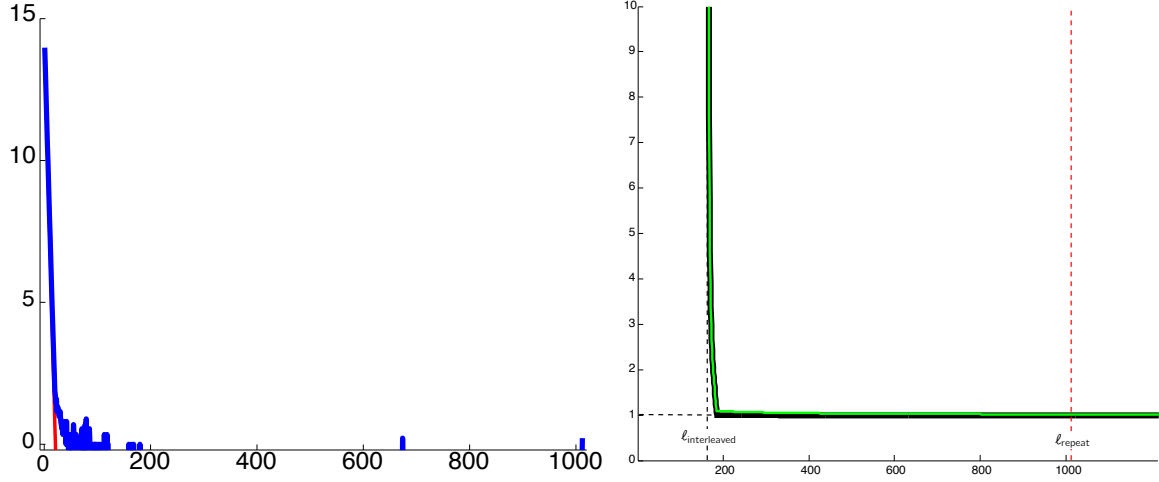


Figure 13: *Salmonella*. $G = 2,215,568$, $\ell_{\text{triple}} = 112$, $\ell_{\text{interleaved}} = 163$, $\ell_{\text{repeat}} = 1011$.

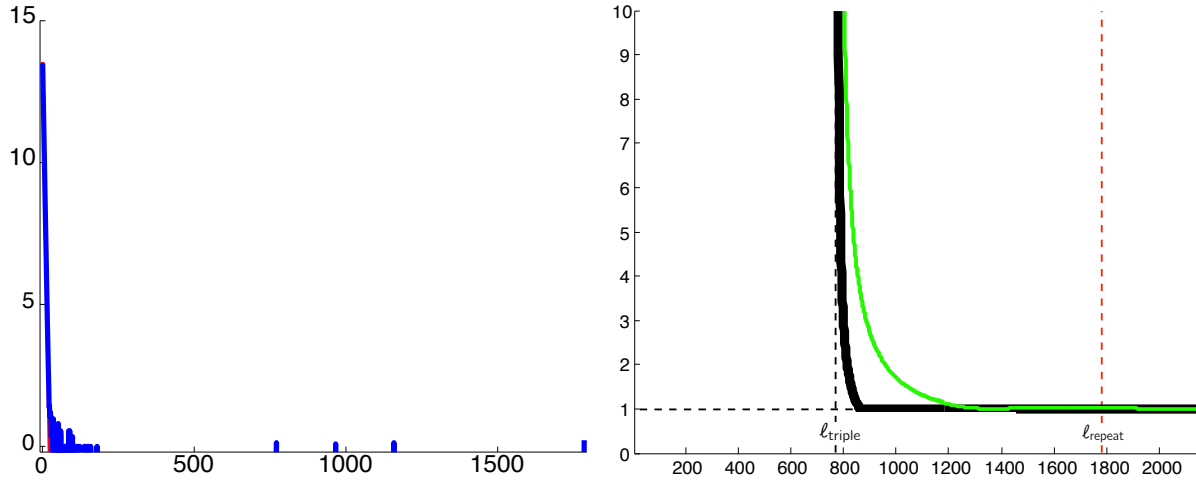


Figure 14: *Perkinsus marinus*. $G = 1,440,372$, $\ell_{\text{triple}} = 770$, $\ell_{\text{interleaved}} = 92$, $\ell_{\text{repeat}} = 1784$.

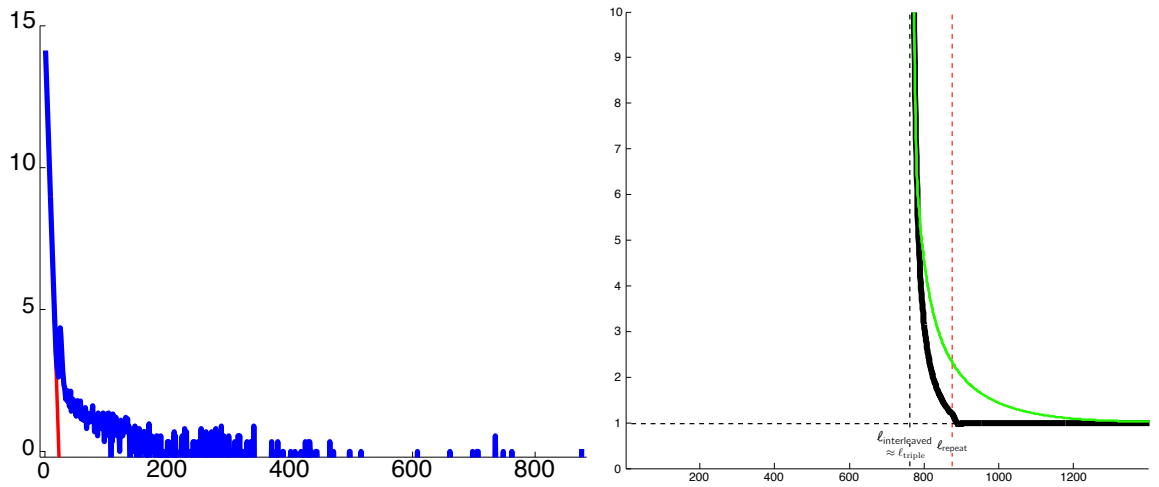


Figure 15: *Sulfolobus islandicus*. $G = 2,655,198$, $\ell_{\text{triple}} = 734$, $\ell_{\text{interleaved}} = 761$, $\ell_{\text{repeat}} = 875$.

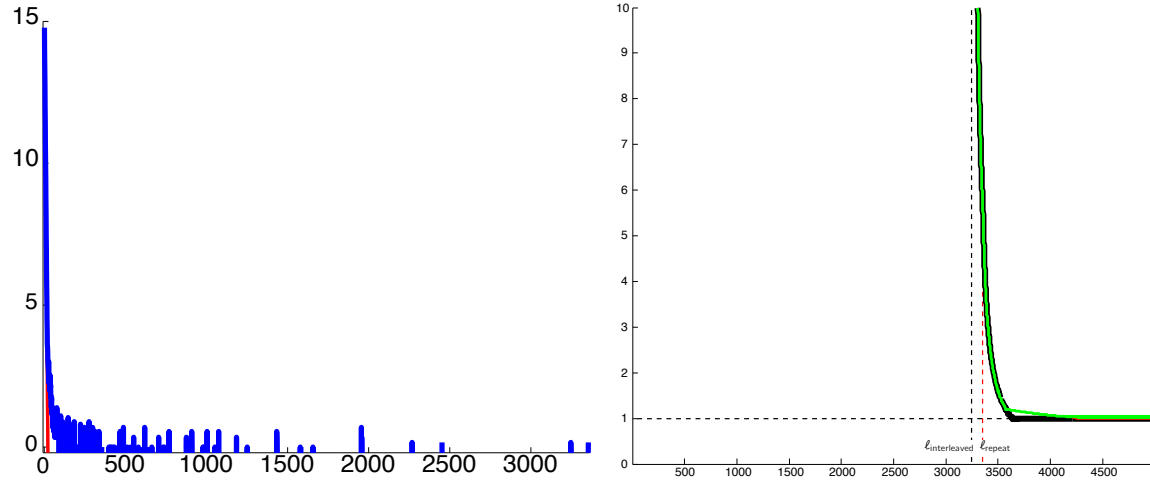


Figure 16: Ecoli536. $G = 4,938,920$, $\ell_{\text{triple}} = 2267$, $\ell_{\text{interleaved}} = 3245$, $\ell_{\text{repeat}} = 3353$.

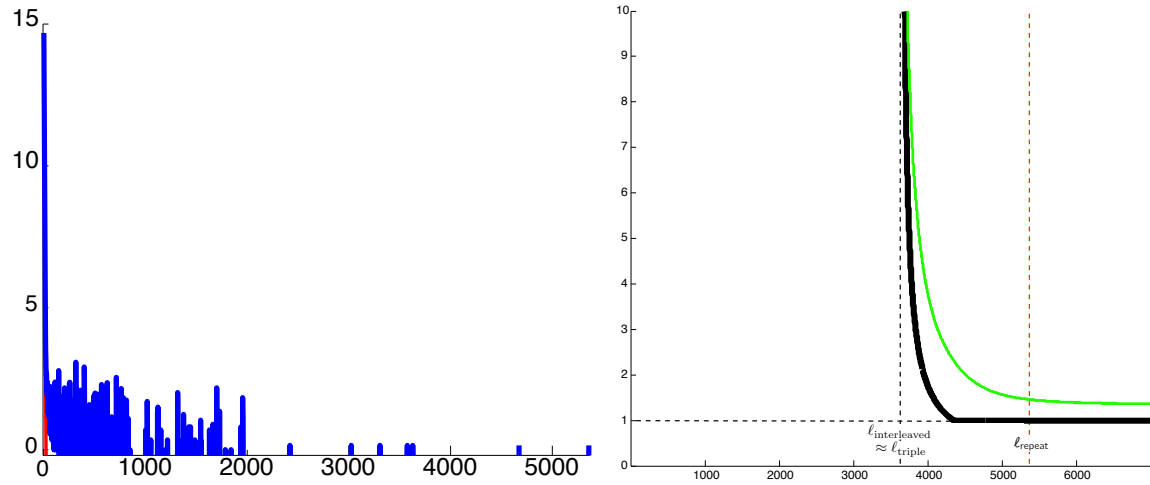


Figure 17: Yesnina. $G = 4,504,254$, $\ell_{\text{triple}} = 3573$, $\ell_{\text{interleaved}} = 3627$, $\ell_{\text{repeat}} = 5358$.

B Lower bounds on coverage depth

The lower bounds are based on a generalization of Ukkonen's condition to shotgun sequencing, as described in Theorem 1. The proof of Theorem 1 follows by a straightforward modification to the argument in [26] and is omitted here.

Theorem 1. *Given a DNA sequence \mathbf{s} and a set of reads, if there is a pair of interleaved repeats or a triple repeat whose copies are all unbridged, then there is another sequence \mathbf{s}' of the same length under which the likelihood of observing the reads is the same.*

B.1 Lower bound due to interleaved repeats

In this section we derive a necessary condition on N and L in order that the probability of correct reconstruction be at least $1 - \epsilon$.

Recall that a pair of repeats, one at positions t_1, t_3 with $t_1 < t_3$ and the second at positions t_2, t_4 with $t_2 < t_4$, is *interleaved* if $t_1 < t_2 < t_3 < t_4$ or $t_2 < t_1 < t_4 < t_3$. From the DNA we may extract a (symmetric) matrix of interleaved repeat statistics b_{mn} , the number of pairs of interleaved repeats of lengths m and n .

We proceed by fixing both N and L and checking whether or not unbridged interleaved repeats occur with probability higher than ϵ . We

will break up repeats into 2 categories: repeats of length at least $L - 1$ (these are always unbridged), and repeats of length less than $L - 1$ (these are sometimes unbridged). We assume that $L > \ell_{\text{interleaved}} + 1$, or equivalently $b_{ij} = 0$ for all $i, j \geq L - 1$, since otherwise there are (with certainty) unbridged interleaved repeats and Ukkonen's condition is violated.

First, we estimate the probability of error due to interleaved repeats of lengths $i < L - 1$ and $j \geq L - 1$. The repeat of length j is too long to be bridged, so an error occurs if the repeat of length i is unbridged. For a repeat, as long as the two copies' locations are not too nearby⁴, each copy is bridged independently and hence the probability that both copies of the repeat of length i are unbridged is $p_i^{\text{unbridged}} = e^{-2\frac{N}{G}(L-i-1)}$. (Recall that a repeat is unbridged if *both* copies are unbridged.)

A union bound estimate⁵ gives a probability of error

$$P_{\text{error}} \approx \frac{1}{2} \sum_{\substack{m < L-1 \\ n \geq L-1}} b_{mn} e^{-2\lambda(L-m-1)}. \quad (8)$$

Requiring the error probability to be less than ϵ and solving for L gives the necessary condition

$$L \geq \frac{1}{2\lambda} \log \frac{\gamma_1}{2\epsilon} = \frac{G}{2N} \log \frac{\gamma_1}{2\epsilon}, \quad (9)$$

where $\gamma_1 := \sum_{\substack{m < L-1 \\ n \geq L-1}} b_{mn} e^{2(N/G)(m+1)}$ is a simple function of the interleaved repeat statistic b_{mn} .

We now estimate the probability of error due to interleaved repeat pairs in which both repeats are shorter than $L - 1$. In this case only one repeat of each interleaved repeat pair must be bridged. Again a union bound estimate gives

$$P_{\text{error}} \approx \frac{1}{2} \sum_{m, n < L-1} b_{mn} e^{-2\lambda(L-m-1)} e^{-2\lambda(L-n-1)}.$$

⁴More precisely, for the two copies of a repeat of length ℓ to be bridged independently requires that no single read can bridge them both. This means their locations t and $t + d$ must have separation $d \geq L - \ell - 2$.

⁵The union bound on probabilities gives an *upper bound*, so its use here is only an approximation. To get a rigorous lower bound we can use the inclusion-exclusion principle, but the difference in the two computations is negligible for the data we observed. For ease of exposition we opt to present the simpler union bound estimate.

Requiring the error probability to be less than ϵ gives the necessary condition

$$L \geq \frac{1}{4\lambda} \log \frac{\gamma_2}{2\epsilon} = \frac{G}{4N} \log \frac{\gamma_2}{2\epsilon}, \quad (10)$$

where $\gamma_2 := \sum_{m,n < L-1} b_{mn} e^{2(N/G)(m+n+2)}$ and similarly to γ_1 is computed from b_{mn} .

B.2 Lower bound due to triple repeats

We translate the generalized Ukkonen's condition prohibiting unbridged triple repeats into a condition on L and N . Let c_m denote the number of triple repeats of length m . Then a union bound estimate gives

$$\mathbb{P}(\mathcal{E}) \approx \frac{1}{2} \sum_m c_m e^{-3\lambda(L-m-1)}. \quad (11)$$

Requiring $\mathbb{P}(\mathcal{E}) \leq \epsilon$ and solving for L gives

$$L \geq \frac{1}{3\lambda} \log \frac{\gamma_3}{2\epsilon} = \frac{G}{3N} \log \frac{\gamma_3}{2\epsilon}, \quad (12)$$

where $\gamma_3 := \sum_m c_m e^{3(N/G)(m+1)}$.

Remark 7. As discussed here and in Section 2, if the DNA sequence is not covered by the reads or there are unbridged interleaved or triple repeats, then reconstruction is not possible. But there is another situation which must be ruled out. Without knowing its length a priori, it is impossible to know how many copies of the DNA sequence are actually present: if the sequence \mathbf{s} to be assembled consists of multiple concatenated copies of a shorter sequence, rather than just one copy, the probability of observing any set of reads will be the same. Since it is unlikely that a true DNA sequence will consist of the same sequence repeated multiple times, we assume this is not the case throughout the paper. Equivalently, if \mathbf{s} does consist of multiple concatenated copies of a shorter sequence, we are content to reconstruct a single copy. If available, knowledge of the approximate length of \mathbf{s} would then allow to reconstruct.

C Proofs for algorithms

C.1 Proof of Theorem 2 (Greedy)

The greedy algorithm's underlying data structure is the overlap graph, where each node represents a read and each (directed) edge (\mathbf{x}, \mathbf{y}) is labeled with the overlap $\text{ov}(\mathbf{x}, \mathbf{y})$ (defined as the length of the shared prefix/suffix) between the incident nodes' reads. For a node \mathbf{v} , the in-degree [out-degree] is the number of edges in the graph directed towards [away from] \mathbf{v} . The greedy algorithm is described as follows.

Algorithm 2 GREEDY. Input: reads \mathcal{R} . Output: sequence $\hat{\mathbf{s}}$.

1. For each read with sequence \mathbf{x} , form a node with label \mathbf{x} .

Greedy steps 2-3:

2. Consider all pairs of nodes $\mathbf{x}_1, \mathbf{x}_2$ in \mathbb{G} satisfying $d_{\text{out}}(\mathbf{x}_1) = d_{\text{in}}(\mathbf{x}_2) = 0$, and add an edge $(\mathbf{x}_1, \mathbf{x}_2)$ with largest value $\text{ov}(\mathbf{x}_1, \mathbf{x}_2)$.

3. Repeat Step 2 until no candidate pair of nodes remains.

Finishing step:

4. Output the sequence corresponding to the unique cycle in \mathbb{G} .

Theorem 2. *Given a sequence \mathbf{s} and a set of reads, GREEDY returns \mathbf{s} if every repeat is bridged.*

Proof. We prove the contrapositive. Suppose GREEDY makes its first error in merging reads \mathbf{r}_i and \mathbf{r}_j with overlap $\text{ov}(\mathbf{r}_i, \mathbf{r}_j) = \ell$. Now, if \mathbf{r}_j is the successor to \mathbf{r}_i , then the error is due to incorrectly aligning the reads; the other case is that \mathbf{r}_j is not the successor of \mathbf{r}_i . In the first case, the subsequence $\mathbf{s}_{t_j}^\ell$ is repeated at location $\mathbf{s}_{t_i+L-\ell}^\ell$, and no read bridges either repeat copy.

In the second case, there is a repeat $\mathbf{s}_{t_j}^\ell = \mathbf{s}_{t_i+L-\ell}^\ell$. If $\mathbf{s}_{t_i+L-\ell}^\ell$ is bridged by some read \mathbf{r}_k , then \mathbf{r}_i has overlap at least $\ell + 1$ with \mathbf{r}_k , implying that read \mathbf{r}_i has already found its successor before step ℓ (either \mathbf{r}_k or some other read with even higher overlap). A similar argument shows that $\mathbf{s}_{t_j}^\ell$ cannot be bridged, hence there is an unbridged repeat. \square

C.2 Proofs for K -mer algorithms

C.2.1 Background

We give some mathematical background leading to the proof of Theorem 3 (restated below).

Lemma 8. *Fix an arbitrary K and form the K -mer graph from the $(K+1)$ -spectrum \mathcal{S}_{K+1} . The sequence \mathbf{s} corresponds to a unique cycle $\mathcal{C}(\mathbf{s})$ traversing each edge at least once.*

To prove the lemma, note that all $(K+1)$ -mers in \mathbf{s} correspond to edges and adjacent $(K+1)$ -mers in \mathbf{s} are represented by adjacent edges. An induction argument shows that \mathbf{s} corresponds to a cycle. The cycle traverses all the edges, since each edge represents a unique $(K+1)$ -mer in \mathbf{s} .

In both SBH and shotgun sequencing the number of times each edge e is traversed by $\mathcal{C}(\mathbf{s})$ (henceforth called the *multiplicity* of e) is unknown a priori, and finding this number is part of the reconstruction task. Repeated $(K+1)$ -mers in \mathbf{s} correspond to edges in the K -mer graph traversed more than once by $\mathcal{C}(\mathbf{s})$, i.e. having multiplicity greater than one. In order to estimate the multiplicity, previous works seek a solution to the so-called Chinese Postman Problem (CPP), in which the goal is to find a cycle of the shortest total length traversing every edge in the graph (see e.g. [20], [6], [22], [11]). It is not obvious under what conditions the CPP solution *correctly* assigns multiplicities in agreement with $\mathcal{C}(\mathbf{s})$. For our purposes, as we will see in Theorem 3, the multiplicity estimation problem can be sidestepped (thereby avoiding solving CPP) through a modification to the K -mer graph.

Ignoring the issue of edge multiplicities for a moment, Pevzner [21] showed for the SBH model that if the edge multiplicities are known with multiple copies of each edge included according to the multiplicities, and moreover Ukkonen’s condition is satisfied, then there is a *unique Eulerian cycle* in the K -mer graph and the Eulerian cycle corresponds to the original sequence. (An Eulerian cycle is a cycle traversing each edge exactly once.) Pevzner’s algorithm is thus to find an Eulerian cycle and read off the corresponding sequence. Both steps can be done efficiently.

Lemma 9 (Pevzner [21]). *In the SBH setting, if the edge multiplicities are known, then there is a unique Eulerian cycle in the K -mer graph with $K = L - 1$ if and only if there are no unbridged interleaved repeats or unbridged triple repeats.*

Most practical algorithms (e.g. [6], [10], [29]) condense unambiguous paths (called unitigs by Myers [15] in a slightly different setting) for computational efficiency. The more significant benefit for us, as shown in Theorem 3, is that if Ukkonen’s condition is satisfied then condensing the graph obviates the need to estimate multiplicities. Condensing a K -mer graph results in a graph of the following type.

Definition 10 (Sequence graph). A *sequence graph* is a graph in which each node is labeled with a subsequence, and edges (\mathbf{u}, \mathbf{v}) are labeled with an *overlap* $a_{\mathbf{uv}}$ such the subsequences \mathbf{u} and \mathbf{v} overlap by $a_{\mathbf{uv}}$ (the overlap is not necessarily maximal). In other words, an edge label $a_{\mathbf{uv}}$ on $e = (\mathbf{u}, \mathbf{v})$ indicates that the $a_{\mathbf{uv}}$ -length suffix of \mathbf{u} is equal to the $a_{\mathbf{uv}}$ -length prefix of \mathbf{v} .

The sequence graph generalizes both the overlap graph used by GREEDY in Section 3.1 (nodes correspond to reads, and edge overlaps are *maximal* overlaps) as well as the K -mer algorithms discussed in this section (nodes correspond to K -mers, and edge overlaps are $K - 1$).

In order to speak concisely about concatenated sequences in the sequence graph, we extend the notation \mathbf{s}_t^ℓ (denoting the length- ℓ subsequence of the DNA sequence \mathbf{s} starting at position t) which was introduced in Section 2.2; we abuse notation slightly, and write $\mathbf{s}_t^{\text{end}}$ to indicate the subsequence of \mathbf{s} starting at position t and having length so that its end coincides with the end of \mathbf{s} .

We will perform two basic operations on the sequence graph. For an edge $e = (\mathbf{u}, \mathbf{v})$ with overlap $a_{\mathbf{uv}}$, *merging* \mathbf{u} and \mathbf{v} along e produces the concatenation $\mathbf{u}_1^{\text{end}} \mathbf{v}_{a_{\mathbf{uv}}+1}^{\text{end}}$. *Contracting* an edge $e = (\mathbf{u}, \mathbf{v})$ entails two steps (c.f. Fig. 6): first, merging \mathbf{u} and \mathbf{v} along e to form a new node $\mathbf{w} = \mathbf{u}_1^{\text{end}} \mathbf{v}_{a_{\mathbf{uv}}+1}^{\text{end}}$, and, second, edges to \mathbf{u} are replaced with edges to \mathbf{w} , and edges from \mathbf{v} are replaced by edges from \mathbf{w} . We will only contract edges (\mathbf{u}, \mathbf{v}) with $d_{\text{out}}(\mathbf{u}) = d_{\text{in}}(\mathbf{v}) = 1$.

The condensed graph is defined next.

Definition 11 (Condensed sequence graph). The *condensed sequence graph* replaces unambiguous paths by single nodes. Concretely, any edge $e = (u, v)$ with $d_{\text{out}}(u) = d_{\text{in}}(v) = 1$ is contracted, and this is repeated until no candidate edges remain.

For a path $\mathcal{P} = \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_q$ in the original graph, the corresponding path in the condensed graph is obtained by contracting an edge $(\mathbf{v}_i, \mathbf{v}_{i+1})$ whenever it is contracted in the graph, replacing the node \mathbf{v}_1 by \mathbf{w} whenever an edge $(\mathbf{u}, \mathbf{v}_1)$ is contracted to form \mathbf{w} , and similarly for the final node \mathbf{v}_q . It is impossible for an intermediate node \mathbf{v}_i , $2 \leq i < q$, to be merged with a node outside of \mathcal{P} , as this would violate the condition $d_{\text{out}}(u) = d_{\text{in}}(v) = 1$ for edge contraction in Defn. 11.

In the condensed sequence graph \mathbb{G} obtained from a sequence \mathbf{s} , nodes correspond to subsequences via their labels, and paths in \mathbb{G} correspond to subsequences in \mathbf{s} via merging the constituent nodes along the path. If the subsequence corresponding to a node \mathbf{v} appears twice or more in \mathbf{s} , we say that \mathbf{v} corresponds to a repeat. Conversely, subsequences of length $\ell \geq K$ in \mathbf{s} correspond to paths \mathcal{P} of length $\ell - K + 1$ in the K -mer graph, and thus by the previous paragraph also to paths in the condensed graph \mathbb{G} .

We record a few simple facts about the condensed sequence graph obtained from a K -mer graph.

Lemma 12. *Let \mathbb{G}_0 be the K -mer graph constructed from the $(K + 1)$ -spectrum of \mathbf{s} and let $\mathcal{C}_0 = \mathcal{C}_0(\mathbf{s})$ be the cycle corresponding to \mathbf{s} . In the condensed graph \mathbb{G} , let \mathcal{C} be the cycle obtained from \mathcal{C}_0 by contracting the same edges as those contracted in \mathbb{G}_0 .*

1. *Edges in \mathbb{G}_0 can be contracted in any order, resulting in the same graph \mathbb{G} , so the condensed graph is well-defined. Similarly \mathcal{C} is well-defined.*
2. *The cycle \mathcal{C} in \mathbb{G} corresponds to \mathbf{s} and is the unique such cycle.*
3. *The cycle \mathcal{C} in \mathbb{G} traverses each edge at least once.*

Theorem 3. *Let \mathcal{S}_{K+1} be the $(K + 1)$ -spectrum of \mathbf{s} and \mathbb{G}_0 be the K -mer graph constructed from \mathcal{S}_{K+1} , and let \mathbb{G} be the condensed sequence graph obtained from \mathbb{G}_0 . If Ukkonen's condition is satisfied, i.e. there are no triple repeats or interleaved repeats of length at least K , then there is a unique Eulerian cycle \mathcal{C} in \mathbb{G} and \mathcal{C} corresponds to \mathbf{s} .*

Proof. We will show that if Ukkonen's condition is satisfied, the cycle $\mathcal{C} = \mathcal{C}(\mathbf{s})$ in \mathbb{G} corresponding to \mathbf{s} (constructed in Lemma 12) traverses each edge exactly once in the condensed K -mer graph, i.e. \mathcal{C} is Eulerian. Pevzner's [21] arguments show that if there are multiple Eulerian cycles then Ukkonen's condition is violated, so it is sufficient to prove that \mathcal{C} is Eulerian. As noted in Lemma 12, \mathcal{C} traverses each edge at least once, and thus it remains only to show that \mathcal{C} traverses each edge *at most* once.

To begin, let \mathcal{C}_0 be the cycle corresponding to \mathbf{s} in the original K -mer graph \mathbb{G}_0 . We argue that every edge (\mathbf{u}, \mathbf{v}) traversed twice by \mathcal{C}_0 in the K -mer graph \mathbb{G}_0 has been contracted in the condensed graph \mathbb{G} and hence in \mathcal{C} . Note that the cycle \mathcal{C}_0 does not traverse any node three times in \mathbb{G}_0 , for this would imply the existence of a triple repeat of length K , violating the hypothesis of the Lemma. It follows that the node \mathbf{u} cannot have two outgoing edges in \mathbb{G}_0 as \mathbf{u} would then be traversed three times; similarly, \mathbf{v} cannot have two incoming edges. Thus $d_{\text{out}}(\mathbf{u}) = d_{\text{in}}(\mathbf{v}) = 1$ and, as prescribed in Defn. 11, the edge (\mathbf{u}, \mathbf{v}) has been contracted. \square

C.2.2 Proofs for SimpleBridging

Since bridging reads extend one base to either end of a repeat, it will be convenient to use the following notation for extending sequences: Given an X-node \mathbf{v} with an incoming edge (\mathbf{p}, \mathbf{v}) and an outgoing edge (\mathbf{v}, \mathbf{q}) , let

$$\mathbf{v}^{\rightarrow \mathbf{q}} = \mathbf{v} \mathbf{q}_{a_{\mathbf{v}\mathbf{q}}+1}^1, \quad \text{and} \quad \mathbf{p}^{\rightarrow \mathbf{v}} = \mathbf{p}_{\text{end}-a_{\mathbf{p}\mathbf{v}}}^1 \mathbf{v}. \quad (13)$$

Here $\mathbf{v}^{\rightarrow \mathbf{q}}$ denotes the subsequence \mathbf{v} appended with the single next base in the merging of \mathbf{v} and \mathbf{q} and $\mathbf{p}^{\rightarrow \mathbf{v}}$ the subsequence \mathbf{v} prepended with the single previous base in the merging of \mathbf{p}

and \mathbf{v} . For example, if $\mathbf{v} = \text{ATTC}$, $\mathbf{p} = \text{TCAT}$, $a_{\mathbf{p}\mathbf{v}} = 2$, $\mathbf{q} = \text{TTCGCC}$, and $a_{\mathbf{v}\mathbf{q}} = 3$, then $\mathbf{v} \rightarrow \mathbf{q} = \text{ATTCG}$, $\mathbf{p} \rightarrow \mathbf{v} = \text{CATTC}$, and $\mathbf{p} \rightarrow \mathbf{v} \rightarrow \mathbf{q} = \text{CATTCG}$.

The idea is that a bridging read is consistent with only one pair $\mathbf{p} \rightarrow \mathbf{v}$ and $\mathbf{v} \rightarrow \mathbf{q}$ and thus allows to match up edge (\mathbf{p}, \mathbf{v}) with (\mathbf{v}, \mathbf{q}) . This is recorded in the following lemma.

Lemma 13. *Suppose \mathcal{C} corresponds to a sequence \mathbf{s} in a condensed sequence graph \mathbb{G} . If a read \mathbf{r} bridges an X-node \mathbf{v} , then there are unique edges (\mathbf{p}, \mathbf{v}) and (\mathbf{v}, \mathbf{q}) such that $\mathbf{p} \rightarrow \mathbf{v}$ and $\mathbf{v} \rightarrow \mathbf{q}$ are adjacent in \mathbf{r} .*

SIMPLEBRIDGING is described as follows.

Algorithm 3 SIMPLEBRIDGING. Input: reads \mathcal{R} , parameter K . Output: sequence $\hat{\mathbf{s}}$.

K-mer steps 1-3:

1. For each subsequence \mathbf{x} of length K in a read, form a node with label \mathbf{x} .
2. For each read, add edges between nodes representing adjacent K -mers in the read.
3. Condense the graph as described in Defn. 11.
4. *Bridging step:* See Fig. 7. While there exists an X-node \mathbf{v} with $d_{\text{in}}(\mathbf{v}) = d_{\text{out}}(\mathbf{v}) = 2$ bridged by some read \mathbf{r} : (i) Remove \mathbf{v} and edges incident to it. Add duplicate nodes $\mathbf{v}_1, \mathbf{v}_2$. (ii) Choose the unique \mathbf{p}_i and \mathbf{q}_j s.t. $\mathbf{p}_i \rightarrow \mathbf{v}$ and $\mathbf{v} \rightarrow \mathbf{q}_j$ are adjacent in \mathbf{r} and add edges $(\mathbf{p}_i, \mathbf{v}_1)$ and $(\mathbf{v}_1, \mathbf{q}_j)$. Choose the unused \mathbf{p}_i and \mathbf{q}_j , add edges $(\mathbf{p}_i, \mathbf{v}_2)$ and $(\mathbf{v}_2, \mathbf{q}_j)$. (iii) Condense the graph.
5. *Finishing step:* Find an Eulerian cycle in the graph and return the corresponding sequence.

C.2.3 Proofs for MultiBridging

In this subsection we recall Theorem 6 stating sufficient conditions for correct reconstruction, and derive the corresponding required coverage depth and read length to meet a target probability of correct reconstruction. The subsection concludes with a proof that the sufficient conditions are correct.

Theorem 6. *The algorithm MULTIBRIDGING reconstructs the sequence \mathbf{s} if:*

- (a) *all interleaved repeats are bridged*

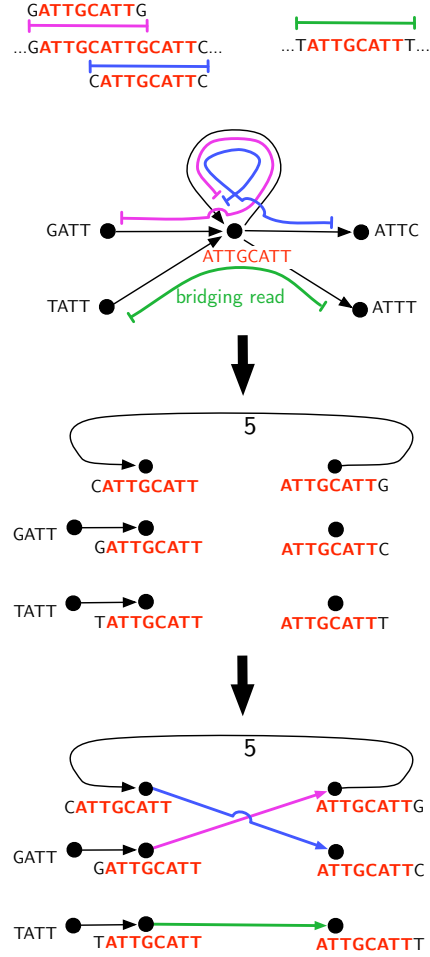


Figure 18: Resolution of X-node with a self-loop.

- (b) *all triple repeats are all-bridged*
(c) *the sequence is covered by the reads.*

Remark 14. Unlike the previous K -mer algorithms, DEBRUIJN and SIMPLEBRIDGING, it is unnecessary to specify a parameter K for MULTIBRIDGING. Implicitly MULTIBRIDGING uses $K = 1$, which makes the condition that reads overlap by K equivalent to coverage of the genome.

Figure 2 plots the performance of MULTIBRIDGING, obtained by solving for the relationship between G, N, L , and ϵ in order to satisfy the conditions of Lemma 6. We first perform the requisite calculations, and then prove the Lemma.

Condition (a) is already dealt with in (9) and (10), and Condition (c) amounts to the requirement that $\frac{N}{N_{\text{LW}}} \geq 1$.

We turn to Condition (b) that all triple repeats are all-bridged. Let c_m denote the number of triple repeats of length m . A union bound estimate over triple repeats for the event that one such triple repeat fails to be all-bridged gives

$$P_{\text{error}} \approx \sum_m 3 \cdot c_m e^{-\lambda(L-m-1)^+}, \quad (14)$$

and requiring $P_{\text{error}} \leq \epsilon$ and solving for L yields

$$L \geq \frac{1}{\lambda} \log \frac{\gamma_3}{\epsilon} = \frac{G}{N} \log \frac{\gamma_3}{\epsilon}, \quad (15)$$

where $\gamma_3 := \sum_m 3c_m e^{(N/G) \cdot (m+1)}$ is computed from the triple repeat statistics c_m .

In order to understand the cost of all-bridging triple repeats, compared to simply bridging one copy as required by our lower bound, it is instructive to study the effect of the single longest triple repeat. Setting $c_{\ell_{\text{triple}}} = 1$ and $c_m = 0$ for $m \neq \ell_{\text{triple}}$ makes $\gamma_3 = 3e^{(N/G) \cdot (\ell_{\text{triple}}+1)}$ in (15) and

$$L \geq L_3^{\text{all}} := \ell_{\text{triple}} + 1 + \frac{G}{N} \log 3\epsilon^{-1}. \quad (16)$$

Bridging the longest triple repeat, as shown in Section B.2, requires

$$L \geq L_3 := \ell_{\text{triple}} + 1 + \frac{G}{3N} \log \epsilon^{-1}. \quad (17)$$

Solving for N in equations (17) and (16) gives

$$N_3 \geq \frac{G}{3} \cdot \frac{\log \epsilon^{-1}}{L - \ell_{\text{triple}} - 1} \quad (18)$$

$$N_3^{\text{all}} \geq G \cdot \frac{\log \epsilon^{-1} + \log 3}{L - \ell_{\text{triple}} - 1}. \quad (19)$$

The ratio is

$$\frac{N_3^{\text{all}}}{N_3} = 3 \cdot \frac{\log 3\epsilon^{-1}}{\log \epsilon^{-1}} \approx 3.72 \quad \text{for } \epsilon = 10^{-2}. \quad (20)$$

This means that if the longest triple repeat is dominant, then for L slightly larger than ℓ_{triple} , MULTIBRIDGING needs a coverage depth approximately 3.72 times higher than required by our lower bound.

The remainder of this subsection is devoted to the proving Lemma 6.

We will use $m_{\mathcal{C}}(\mathbf{v})$ to denote the multiplicity (traversal count) a cycle \mathcal{C} assigns a node \mathbf{v} . The multiplicity $m_{\mathcal{C}}(\mathbf{v})$ is also equal to the number of times the subsequence \mathbf{v} appears in the sequence corresponding to \mathcal{C} . For an edge e , we can similarly let $m_{\mathcal{C}}(e)$ be the number of times \mathcal{C} traverses the edge. The following key lemma relates node multiplicities with the existence of X-nodes.

Lemma 15. *Let \mathcal{C} be a cycle in a condensed sequence graph \mathbb{G} , where \mathbb{G} itself is not a cycle, traversing every edge at least once. If \mathbf{v} is a node with maximum multiplicity at least 2, i.e. $m_{\mathcal{C}}(\mathbf{v}) = \max_{\mathbf{u} \in \mathbb{G}} m_{\mathcal{C}}(\mathbf{u}) \geq 2$, then \mathbf{v} is an X-node. As a consequence, if $m_{\mathcal{C}}(\mathbf{v}) \geq 3$ for some \mathbf{v} , i.e. \mathcal{C} traverses some node at least three times, then $m_{\mathcal{C}}(\mathbf{u}) \geq 3$ for some X-node \mathbf{u} .*

Proof. Let \mathbf{v} be a node with maximum multiplicity $m_{\mathcal{C}}(\mathbf{v}) = \max_{\mathbf{u} \in \mathbb{G}} m_{\mathcal{C}}(\mathbf{u})$. We will show that \mathbf{v} is an X-node, i.e. $d_{\text{out}}(\mathbf{v}) \geq 2$ and $d_{\text{in}}(\mathbf{v}) \geq 2$.

We prove that $d_{\text{out}}(\mathbf{v}) \geq 2$ by supposing that $d_{\text{out}}(\mathbf{v}) = 1$ and deriving a contradiction. Denote the outgoing edge from \mathbf{v} by $e = (\mathbf{v}, \mathbf{u})$, where \mathbf{u} is distinct from \mathbf{v} since otherwise \mathbb{G} is a cycle. If $d_{\text{in}}(\mathbf{u}) \geq 2$, then \mathbf{u} must be traversed more times than \mathbf{v} , contradicting the maximality of $m_{\mathcal{C}}(\mathbf{v})$, and if $d_{\text{in}}(\mathbf{u}) = 1$, then the existence of the edge e contradicts the fact that \mathbb{G} is condensed. The argument showing that $d_{\text{in}}(\mathbf{v}) \geq 2$ is symmetric to the case $d_{\text{in}}(\mathbf{v}) \geq 2$. \square

Proof of Lemma 6. We assume that all triple repeats are all-bridged, that there are no unbridged interleaved repeats, and that all reads overlap their successors by at least 1 base pair. We wish to show that MULTIBRIDGING returns the original sequence.

Consider the condensed sequence graph \mathbb{G}_0 constructed in steps 1-3 of MULTIBRIDGING. Suppose all X-nodes that are either all-bridged or correspond to bridged 2-repeats have been resolved according to repeated application of the procedure in step 4 of MULTIBRIDGING, resulting in a condensed sequence graph \mathbb{G} . We claim that 1) \mathbf{s} corresponds to a cycle \mathcal{C} in \mathbb{G} traversing every edge at least once, 2) \mathcal{C} is Eulerian, and 3) \mathcal{C} is the unique Eulerian cycle in \mathbb{G} .

Proof of Claim 1. Let \mathbb{G}_n be the graph after n resolution steps, and suppose that \mathcal{C}_n is a cycle in \mathbb{G}_n corresponding to the sequence \mathbf{s} and traversing all edges. We will show that there exists a cycle \mathcal{C}_{n+1} in \mathbb{G}_{n+1} corresponding to \mathbf{s} and traversing all edges, and that $\mathbb{G}_t = \mathbb{G}$ for a finite t , so by induction, there exists a cycle \mathcal{C} in \mathbb{G} corresponding to \mathbf{s} and traversing all edges. The base case $n = 0$ was shown in Lemma 8. Moving on to arbitrary $n > 0$, let \mathbf{v} be an X-node in \mathbb{G}_n labeled as in Fig. 6. The X-node resolution step is constructed precisely to preserve the existence of a cycle corresponding to \mathbf{s} . Each traversal of \mathbf{v} by the cycle \mathcal{C}_n assigns an incoming edge $(\mathbf{p}_i, \mathbf{v})$ to an outgoing edge $(\mathbf{v}, \mathbf{q}_j)$, and the resolution step correctly determines this pairing by the assumption on bridging reads.

Note that all X-nodes in the graph \mathbb{G}_{n+1} continue to correspond to repeats in \mathbf{s} . The process terminates: let $\mathcal{L}(\mathbb{G}_i) = \sum_{\mathbf{v} \in \mathbb{G}_i} m_{\mathcal{C}_i}(\mathbf{v}) \mathbf{1}_{m_{\mathcal{C}_i}(\mathbf{v}) > 1}$ and observe that $\mathcal{L}(\mathbb{G}_i)$ is strictly decreasing in i . Thus \mathbf{s} corresponds to a cycle \mathcal{C} in \mathbb{G} traversing each edge at least once.

Proof of Claim 2. We next show that \mathcal{C} is an Eulerian cycle. If \mathbb{G} is itself a cycle, and \mathbf{s} is not formed by concatenating multiple copies of a shorter subsequence (assumed not to be the case, see discussion at end of Section 2), then \mathcal{C} traverses \mathbb{G} exactly once and is an Eulerian cycle. Otherwise, if \mathbb{G} is not a cycle, then we may apply Lemma 15 to see that any node with $m_{\mathcal{C}}(\mathbf{v}) \geq 3$ implies the existence of an X-node \mathbf{u} with $m_{\mathcal{C}}(\mathbf{u}) \geq 3$. Node \mathbf{u} must be all-bridged, by hypothesis, which means that an additional X-node resolution step can be applied to \mathbb{G} , a contradiction. Thus each node \mathbf{v} in \mathbb{G} has multiplicity $m_{\mathcal{C}}(\mathbf{v}) \leq 2$.

We can now argue that no edge $e = (\mathbf{u}, \mathbf{v})$ is traversed twice by \mathcal{C} in the condensed sequence graph \mathbb{G} , as it would have been contracted. Suppose $m_{\mathcal{C}}(e) \geq 2$. The node \mathbf{u} cannot have two outgoing edges as this implies $m_{\mathcal{C}}(\mathbf{u}) \geq 3$; similarly, \mathbf{v} cannot have two incoming edges. Thus $d_{\text{out}}(\mathbf{u}) = d_{\text{in}}(\mathbf{v}) = 1$, but by Defn. 11 the edge $e = (\mathbf{u}, \mathbf{v})$ would have been contracted.

Proof of Claim 3. It remains to show that there is a *unique* Eulerian cycle in \mathbb{G} . All X-nodes in \mathbb{G} must be unbridged 2-X-nodes (correspond to 2-repeats in \mathbf{s}), as all other X-nodes were assumed to be bridged and have thus been resolved in \mathbb{G} .

We will map the sequence \mathbf{s} to another sequence \mathbf{s}' , allowing us to use the characterization of Lemma 9 for SBH with known multiplicities. Denote by \mathbb{G}' the graph obtained by relabeling each node in \mathbb{G} by a single unique symbol (no matter the original node label length), and setting all edge overlaps to 0. Through the relabeling, \mathcal{C} corresponds to a cycle \mathcal{C}' in \mathbb{G}' , and let \mathbf{s}' be the sequence corresponding to \mathcal{C}' . Writing \mathcal{S}'_2 for the 2-spectrum of \mathbf{s}' , the graph \mathbb{G}' is by construction precisely the 1-mer graph created from \mathcal{S}'_2 , and there is a one-to-one correspondence between X-nodes in \mathbb{G}' and unbridged repeats in \mathbf{s}' . Through the described mapping, every unbridged repeat in \mathbf{s}' maps to an unbridged repeat in \mathbf{s} , with the *order of repeats preserved*.

There are multiple Eulerian cycles in \mathbb{G} only if there are multiple Eulerian cycles in \mathbb{G}' since the graphs have the same topology, and by Lemma 9 the latter occurs only if there are unbridged interleaved repeats in \mathbf{s}' , which by the correspondence in the previous paragraph implies the existence of unbridged interleaved repeats in \mathbf{s} . \square

C.3 Truncation estimate for bridging repeats (GREEDY and MULTI-BRIDGING)

The repeat statistics a_m and c_m used in the algorithm performance curves are potentially overestimates. This is because a large repeat family—one with a large number of copies f —will result in a contribution $\binom{f}{2} \approx f^2/2$ to a_m and $\binom{f}{3} \approx f^3/6$ to c_m .

We focus here on deriving an estimate for the required N, L for bridging all repeats with probability $1 - \epsilon$. This upper bound reduces the sensitivity to large families of short repeats. The analogous derivation for all-bridging all triple-repeats is very similar and is omitted.

Suppose there are a_m repeats of length m . The probability that some repeat is unbridged is ap-

proximately, by the union bound estimate,

$$\mathbb{P}(\mathcal{E}) \approx \sum_m a_m e^{-2\lambda(L-m)}. \quad (21)$$

Requiring $\mathbb{P}(\mathcal{E}) \leq \epsilon$ and solving for L gives

$$L \geq \frac{1}{2\lambda} \log \frac{\gamma}{\epsilon} = \frac{G}{2N} \log \frac{\gamma}{\epsilon}, \quad (22)$$

where $\gamma := \sum_m a_m e^{2(N/G)m}$. Now, if a_m *overcounts* the number of repeats for small values of m , the bound in (22) might be loose. In order for each read to overlap the subsequent read by x nucleotides, with probability of failure $\epsilon/2$, it suffices to take

$$L \geq L_{\text{K-cov}}\left(x, \frac{\epsilon}{2}\right) := x + \frac{1}{\lambda} \log \frac{2N}{\epsilon}. \quad (23)$$

Thus, for any $x < L$, we may replace (22) by

$$L \geq \min_x \max\left\{\frac{1}{2\lambda} \log \frac{2\gamma(x)}{\epsilon}, L_{\text{K-cov}}\left(x, \frac{\epsilon}{2}\right)\right\}, \quad (24)$$

where $\gamma(x) = \sum_{m>x} a_m e^{2(N/G)m}$, and obtain a *looser* bound.

D Critical window calculations

D.1 Window size if $\ell_{\text{interleaved}} \gg \ell_{\text{triple}}$

We focus here on the bound due to interleaved repeats (rather than triple repeats, treated subsequently), and furthermore assume that the effect of the single largest interleaved repeat is dominant. In this case $\ell_{\text{interleaved}} = L_{\text{crit}} - 1$ is the length of the shorter of the pair of interleaved repeats, and let ℓ_1 be the length of the longer of the two. For $L_{\text{crit}} < L \leq \ell_1 + 1$, we are in the setting of (9) but with a redefined $\gamma_1 = e^{2(N/G)(L_{\text{crit}}-1)}$. Thus,

$$L \geq L_{\text{crit}} + \frac{G}{2N} \log \epsilon^{-1}, \quad (25)$$

and solving for N gives

$$N_{\text{repeat}} = \frac{G}{2} \frac{\log \epsilon^{-1}}{L - \ell_2 - 1} \quad (26)$$

Let L^* be the value of L at which the curve described by constraint (26) intersects the Lander-Waterman coverage value, i.e. $N_{\text{repeat}}(L^*) =$

$N_{\text{LW}}(L^*) := N^*$. This is the minimum read length for which coverage of the sequence suffices for reconstruction.

We now solve for $\frac{L^*}{L_{\text{crit}}}$. First, the Lander-Waterman equation (2) at $N = N^*$ is

$$N^* = \frac{G}{L^*} \log \frac{N^*}{\epsilon}, \quad (27)$$

and setting equal the right-hand sides of (27) and (26) at $L = L^*$ gives

$$\frac{G}{L^*} \log \frac{N^*}{\epsilon} = \frac{G}{2} \frac{\log \epsilon^{-1}}{L^* - \ell_2 - 1}.$$

A bit of algebra yields

$$\frac{L^*}{L_{\text{crit}}} = \frac{2}{2-x}, \quad (28)$$

where

$$x := \frac{\log \epsilon^{-1}}{\log N^* + \log \epsilon^{-1}}. \quad (29)$$

Since $x \leq \frac{1}{2}$, equation (28) implies $L^* \leq 2L_{\text{crit}}$, and combined with the obvious inequality $L^* \geq L_{\text{crit}}$, we have $L_{\text{crit}} \leq L^* \leq 2L_{\text{crit}}$. Thus

$$N_{\text{LW}}(2L_{\text{crit}}) \leq N^* \leq N_{\text{LW}}(L_{\text{crit}}), \quad (30)$$

and applying the Lander-Waterman fixed-point equation (2) yet again gives

$$\frac{G}{2L_{\text{crit}}} \log \frac{N_{\text{LW}}(2L_{\text{crit}})}{\epsilon} \leq N^* \leq \frac{G}{L_{\text{crit}}} \log \frac{N_{\text{LW}}(L_{\text{crit}})}{\epsilon}. \quad (31)$$

Writing this out gives

$$\begin{aligned} \frac{\log \epsilon^{-1}}{\log \frac{G}{L_{\text{crit}}} + \log \log \frac{N_{\text{LW}}(L_{\text{crit}})}{\epsilon} + \log \epsilon^{-1}} &\leq x \\ &\leq \frac{\log \epsilon^{-1}}{\log \frac{G}{L_{\text{crit}}} - 1 + \log \log \frac{N_{\text{LW}}(2L_{\text{crit}})}{\epsilon} + \log \epsilon^{-1}}, \end{aligned}$$

and this can be relaxed to

$$\begin{aligned} \frac{\log \epsilon^{-1}}{\log \frac{G}{L_{\text{crit}}} + \log \epsilon^{-1} + \log \log \frac{G}{\epsilon L_{\text{crit}}}} &\leq x \\ &\leq \frac{\log \epsilon^{-1}}{\log \frac{G}{L_{\text{crit}}} - 1 + \log \epsilon^{-1}}. \end{aligned} \quad (32)$$

Letting

$$r := \frac{\log \frac{G}{L_{\text{crit}}}}{\log \epsilon^{-1}}, \quad (33)$$

we have to a very good approximation

$$\frac{L^*}{L_{\text{crit}}} \approx \frac{2(r+1)}{2(r+1)-1}. \quad (34)$$

For $G \sim 10^8$, $L_{\text{crit}} \sim 1000$, and $\epsilon = 5\%$, we get $\log \frac{G}{L_{\text{crit}}} \approx 13.8$ and $\log \epsilon^{-1} \approx 3.0$, so $r \approx 4.6$ and

$$\frac{L^*}{L_{\text{crit}}} = \frac{2(r+1)}{2(r+1)-1} \approx 1.1.$$

From (33) we see that the relative size of $\log \epsilon^{-1}$ and $\log \frac{G}{L_{\text{crit}}}$ determines the size of the critical window. If in the previous example $\epsilon = 10^{-5}$, say, then $\frac{L^*}{L_{\text{crit}}}$ increases to 1.3. As ϵ tends to zero, r approaches zero as well and $\frac{L^*}{L_{\text{crit}}} \rightarrow 2$.

D.2 Window size if $\ell_{\text{triple}} \gg \ell_{\text{interleaved}}$

We now suppose the single longest triple repeat dominates the lower bound and estimate the size of the critical window. In this case $\ell_{\text{triple}} = L_{\text{crit}} - 1$ is the length of the longest triple repeat. Since we don't have matching lower and upper bounds for triple repeats, we separately compute the critical window size for each.

We start with the lower bound. For $L > L_{\text{crit}}$, the minimum value of N required in order to bridge the longest triple repeat is given by (18) and repeated here:

$$N_{\text{triples}} = \frac{G}{3} \cdot \frac{\log \epsilon^{-1}}{L - L_{\text{crit}}}. \quad (35)$$

As for the interleaved repeats case considered earlier, we let L^* be the value of L at which the curve described by constraint (35) intersects the Lander-Waterman coverage value, i.e. $N_{\text{triple}}(L^*) = N_{\text{LW}}(L^*) := N^*$. This is the minimum read length for which coverage of the sequence suffices for reconstruction.

A similar procedure as leading to (28) gives $L^*/L_{\text{crit}} = 3/(3-x)$ with x defined in (29). One can check that the estimates on x in (32) continue to hold, and we therefore get

$$\frac{L^*}{L_{\text{crit}}} \approx \frac{3(r+1)}{3(r+1)-1}. \quad (36)$$

For the same example as before, $G \sim 10^8$, $L_{\text{crit}} \sim 1000$, and $\epsilon = 5\%$, we get $r \approx 4.6$ and

$$\frac{L^*}{L_{\text{crit}}} = \frac{3(r+1)}{3(r+1)-1} \approx 1.06.$$

Changing ϵ to 10^{-5} makes $\frac{L^*}{L_{\text{crit}}} \approx 1.17$, and as ϵ (and hence also r) tends to zero, $\frac{L^*}{L_{\text{crit}}} \rightarrow \frac{3}{2}$.

The analogous computation for L^*/L_{crit} for the upper bound, as given by N_3^{all} in (18), yields

$$\frac{L^*}{L_{\text{crit}}} = \frac{r+1}{r + \frac{\log 3}{\log \epsilon^{-1}}} \approx 1.12, \quad (37)$$

for the example with $G \sim 10^8$, $L_{\text{crit}} \sim 1000$, and $\epsilon = 5\%$. The critical window size of the upper bound is about twice as large as that of the lower bound for typical values of G and L_{crit} , with ϵ moderate. But as $\epsilon \rightarrow 0$, we see from (37) that $L^*/L_{\text{crit}} \rightarrow \infty$, markedly different to the $L^*/L_{\text{crit}} \rightarrow \frac{3}{2}$ observed for the lower bound.